For B.Tech. Bioinformatics Program

FUSION PROTEIN LINKERS THE NEXT FRONTIER OF ONCOLOGICAL RESEARCH

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

BIOINFORMATICS

SUBMITTED BY

SAKSHI PANDEY

UNDER THE GUIDANCE OF

DR. PRASANNA VENKATRAMAN

ADVANCED CENTRE FOR TREATMENT, RESEARCH AND EDUCATION IN CANCER (ACTREC)
TATA MEMORIAL CENTRE

SCHOOL OF BIOTECHNOLOGY AND BIOINFORMATICS

DY PATIL DEEMED TO BE UNIVERSITY, NAVI MUMBAI

JULY, 2020

D Y PATIL UNIVERSITY
NAVI MUMBAI

Sakshi Pandey <sak.pan.bt16@dypatil.edu>

Certificate of completion of Dissertation Training

Venkatraman Prasanna <vprasanna@actrec.gov.in> Thu, Jul 2, 2020 at 1:34 PM
To: sak.pan.bt16@dypatil.edu

I certify that the research work presented in the thesis titled "Fusion Protein Linkers the Next Frontier of Oncological Research" has been carried out by Sakshi Pandey, Roll No. BBI-16005 under my supervision and this is her bona fide work. The research work is original and has not been submitted for any other degree of this or any other university. Further, she was a regular student and has worked under my guidance as a full-time student at Tata Memorial Centre Advanced Centre for Treatment, Research and Education in Cancer, Kharghar, Navi Mumbai until the submission of the thesis to the D.Y. Patil Deemed to be University.

DECLARATION BY THE CANDIDATE

This is to certify that the work embodied in this thesis titled "Fusion Protein Linkers the Next Frontier of Oncological Research" forms my own contribution to the research work carried out under the guidance of Dr. Prasanna Venkatraman at Tata Memorial Centre Advanced Centre for Treatment, Research and Education in Cancer, Kharghar, Navi Mumbai this work has not been submitted for any degree for this University or any other University. Whenever references have been made to previous work of others, it has been clearly indicated as such and included in the Bibliography.

Sakshi Pandey

BBI-16005

Dedication

I would like to dedicate this thesis to my parents and my grandmother who unfalteringly trusted all my decisions, and always allowed me to make the choices I wanted to make. They were always there to offer comfort, support, and wise words to help when the choices I made led me down the wrong path, and always there to cheer for me when they led me to success. Without them none of this would have been possible.

Acknowledgement

I would like to acknowledge my teacher, especially the one who started me on this journey, Ms. Eberhardt without you igniting a spark within me for this subject I would never have got to where I am now.

To all my teachers at DY Patil School of Biotechnology and Bioinformatics I am forever grateful for your guidance and encouragement.

But most importantly I would like to thank Dr. Prasanna Venkatraman, my guide and friend, who offered unwavering support, invaluable teachings, and was available round the clock to answer any queries I might have had (however stupid).

Abstract

Recently there's been a surge of new treatments for cancer, largely thanks to advances in nanotechnology, genetic engineering studies, and data science. Currently, several researchers are focused on the development of cell therapies, anti-tumor vaccines, and new biotechnological drugs that have already shown promising results in preclinical studies. We believe the focus should lie elsewhere, we believe that the future of cancer treatment lies in data. Repeatedly we've seen machines decipher solutions and find patterns where no one has been able to; Their capability to solve problems in an accurate, and unbiased way is an enormous asset that we're going to see utilized in a big way in the coming years. All that's needed is *data*.

The bulk of the current research seems to be focused on finding new cutting edge treatments using novel technologies, with very little research being done on topics which are considered well explored and "known". However there are quite a few lacunae in our understanding of important research areas such as fusion proteins which potentially play a big role in the genesis of cancerous tumors. In our research we found that a distinct component of a fused protein is the junction/linker where the two separate proteins are fused at a breakpoint which is generally found in intrinsically disordered regions.The selection of such a region could indicate that the created fusion proteins are designed to maintain viability and evade degradation pathways. These disordered regions which link together the two proteins fusing them to create oncoproteins seem to have the following properties:they are highly disordered, flexible, hydrophilic residues, and coils. This led us to thinking, what if this disordered linker is a vital functional component in oncoproteins.

Thus we came to the consensus that the "linker" region at the fusion junction seems to hold a lot of interest for future research. If the machine learning model has enough data on these regions it could identify traits, which allow us to distinguish oncoproteins better; Improve our understanding of the linker region's importance, and further aid research into understanding the role of fusion proteins in tumor genesis. We could even derive better treatment options if we can discover a way to use the linker region to inactivate fusion proteins aiding tumor genesis. The possibilities are endless...

Table of Contents

List of Figures

List of Tables

List of Abbreviations

1. IDR: Intrinsically disordered regions
2. IDP: Intrinsically disordered proteins
3. ML: Machine Learning
4. HTML: Hypertext Markup Language
5. PHP: Hypertext Preprocessor; Personal Home Page
6. PANDAS: Python Data Analysis Library
7. NUMPY: Numerical Python
8. SQL: Structured Query Language
9. XAMPP: Cross platform - Apache, MySQL, Perl and PHP
10. Tf: Transferrin
11. hGH: human growth hormone
12. HEK293: Human Embryonic Kidney Cells
13. MICU2: Mitochondrial calcium uptake
14. SH3: SRC Homology 3 Domain
15. DNA: Deoxyribonucleic acid
16. UNIX: Uniplexed Information and Computing Service
17. mRNA: messenger ribonucleic acid
18. FASTA: fast-all
19. BLAST: basic local alignment search tool
20. BLASTp: basic local alignment search tool for proteins
21. CML: chronic myelogenous leukemia
22. MAPK: mitogen-activated protein kinase
23. CSS: Cascading style sheets
24. COSMIC: Catalogue Of Somatic Mutations In Cancer
25. XML: eXtensible Markup Language
26. FLI1: Friend leukemia integration 1 transcription factor
27. EWSR1: EWS RNA Binding Protein
28. ETS: Erythroblast Transformation Specific
29. HSP:A High-scoring Segment Pair

Chapter 1: Introduction

There are over a 100 types of cancer in the world. Of these 20% of deaths occur due to cancers caused by fusion proteins(Mitelman et al. 233). Most, if not all of these fusion proteins are thought to be major players in several deadly cancers. However our knowledge on their sequential and structural features, the way they function, and how they affect tumorigenesis remains relatively limited.

1.1. A Brief History of Cancer:

Historical findings of patients with cancer can first be dated back to ancient Egypt. There is evidence that the Egyptians treated the disease by cutting out the tumor and cauterizing the wound, and that this was often ineffective, and led to the death of patients. Over time we were better able to identify the biological and pathological features of tumors, but a new advance in treatment wasn't made until the discovery of radium. At the end of the 1800s X-rays and their use for the treatment of tumors provided the first modern therapeutic approach in medical oncology. Another major breakthrough took place soon after the Second World War, with the discovery of cytotoxic antitumor drugs and the birth of chemotherapy for the treatment of various hematological and solid tumors. Starting from this epochal turning point, there has been an exponential growth of studies concerning the use of new drugs for cancer treatment. One of the latest areas of innovation is in cancer immunotherapy; These drugs are designed to help the immune system identify and attack cancer cells(Chung and Kim 7).

1.2. What are Fusion Proteins and What is Their Significance in Cancer:

Fusion proteins can be best defined as chimeras. In greek mythology a chimera is a rare creature which is a hybrid mix between a lion, a goat, and a snake. Similarly fusion proteins are rare proteins which consist of sequence regions from different proteins. They are created when two or more genes which are used to code for separate proteins are

altered to create a new fusion gene. Translation of this fusion gene usually results in a polypeptide with functional properties derived from each of the original parent proteins, but in some cases a loss of function can also be observed.

When gene fusions occur they often result in fully functional proteins. Some can experience interactions between the two proteins that can modify their functions; Other gene fusions may cause regulatory changes that can alter when and where these genes act(Starks 94-95). For partial gene fusions, the shuffling of different active sites and binding domains have been observed to demonstrate the potential to result in new proteins with novel functions.Their functions range from signaling, DNA(Deoxyribonucleic acid) polymerization, to primary metabolism(Starks 94-95).

Many fusion genes which occur naturally in our body are thought to be drivers behind cancer, and function as oncogenes in the bodies where they are formed. The proto-oncogene(normal gene) can become an oncogene by a relatively small modification of its original function. (Mertens et al. 371-381) can be summarized as stating that most tumorigenic gene fusions consist of regulatory and protein‑encoding sequences from two different genes and, depending on the fusion gene, one or both the partner genes can contribute to oncogenesis. These fusion genes are known to have several mechanisms of action when it comes to causing tumorigenesis. Examples of pathologically relevant gene fusion events include the juxtaposition of promoter and enhancer sequences close to a proto‑oncogene, the disruption of a tumor suppressor gene, or the creation of a fusion protein with aberrant functionality. There are three basic methods of activation of oncogenes.

According to Lodish H et al. in section (24.2) firstly there could be an increase in the protein concentration due to some kind of misregulation, leading to increased protein expression. The increase could also occur due to messenger ribonucleic acid(mRNA) being more stable, and its longer existence in the cell could lead to an increase in activity. A chromosomal abnormality could also contribute to this phenomenon. Which brings us to the second method, chromosomal translocation. The proto-oncogene could be

translocated such that there is a fusion between it and another gene, which leads to it have greater oncogenic activity; Or a translocation could move the proto-oncogene to a new site on a chromosome leading to higher protein expression. The third method of activation could be a mutation taking place in a regulatory region or within the proto-oncogene itself, changing the protein structure and thus leading to an increase in the enzyme activity and/or loss of regulation. Some examples of proteins which if mutated take part in tumorigenesis, can be observed in Figure 1.

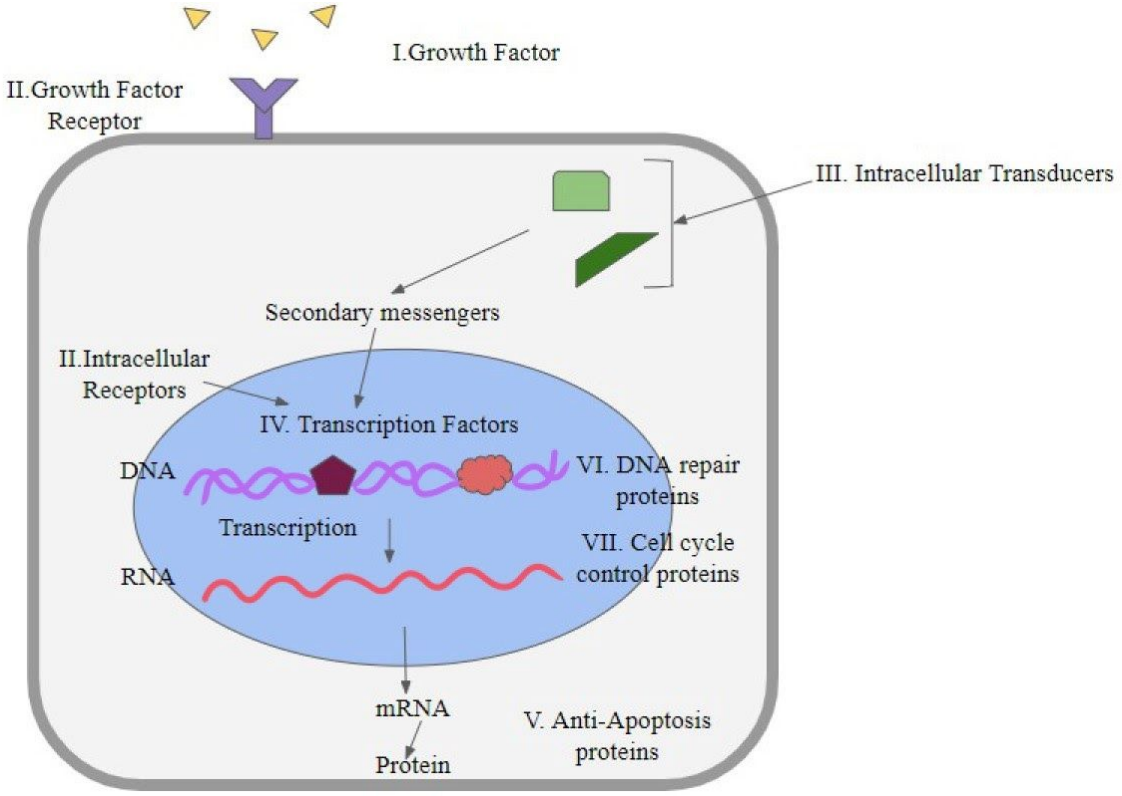


Figure 1. 7 proteins that are vital to controlling cell growth. When the mutant forms of these proteins are expressed, they usually cause cancer.

Fusion proteins are of particular interest in cancer since they are a largely untapped resource that can provide great breakthroughs in diagnostics, and lead to therapeutic advantages. As mentioned above, they play a vital role in tumorigenesis since most of them help to regulate cell growth and differentiation, or they may be involved in signal transduction and execution of mitogenic signals through their protein products. However when we originally discovered fusion proteins they were thought to only play a role in

hematologic cancers, it was only over time that we have come to discover that they play a role in cancers of solid tumors as well. This includes sarcomas, carcinomas, and tumors of the central nervous system.

An example of a fusion protein can be observed in Figure 2 in order to gain better understanding of their creative modes of function.

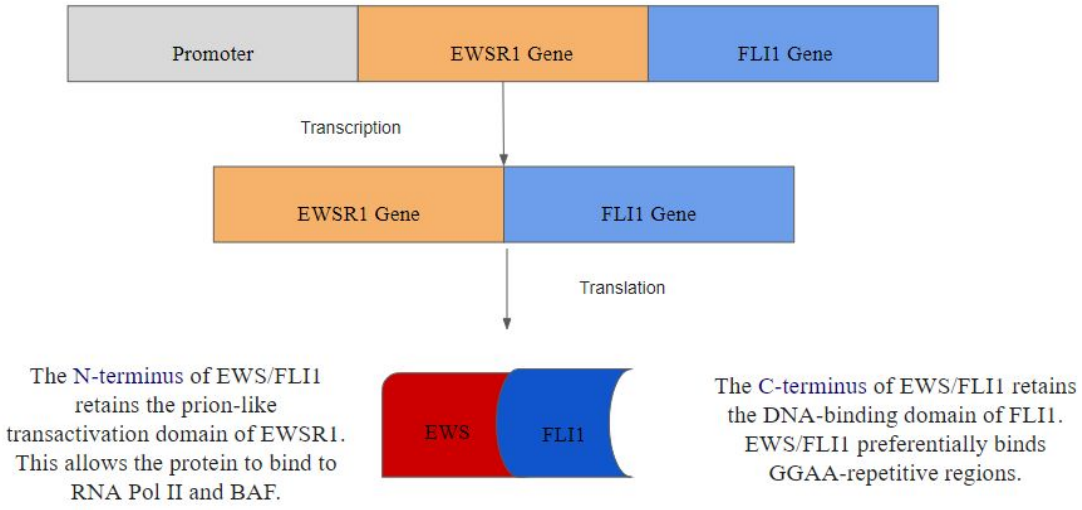


Figure 2. EWS-FLI1 fusion protein and its function. EWS-FLI1 is the most prevalent protein found in a deadly form of cancer called Ewing Sarcoma. The interactions done by EWS/FLI1 change heterochromatin to euchromatin at the sites where the FLI1 domain binds to DNA(Johnson et al. 11-12; Bouley et al. 163). Typically, FLI and other ETS family members bind DNA via their conserved DNA binding domain at the consensus sequence 'ACCGGAAGTG'; In Ewing sarcoma, however, EWS/FLI displays a "gain-of-function" in its ability to also bind 'GGAA'-containing microsatellite (repeat) regions to regulate some of its targets, such as key oncogenic target *NR0B1*(Johnson et al. 11-12). This gain of function allows EWS/FLI1 to bind to a wider range of targets than it normally could, and thus leads to disruption in regulation of normal genes(Riggi et al. 2).

Recent bioinformatics work on gene fusions includes fusion protein domain content and recombination, reading frame conservation, and the protein expression properties. However according to our surveys of various journals there's little to no work being carried out on the sequence and structure relationship of the fusion proteins in regards to the linker at the junction of the two distinct domains which have been fused. We believe this to be a region of interest which may be a crucial design feature in the structure of the fusion protein.

1.3. Background & The Problem to be Investigated:

Oncoproteins putatively seem to have a close relationship with tumorigenesis. One current problem is that even though we know that they play an important role, we are yet to understand the exact mechanism which leads to tumors. The molecular functions of gene fusions, and the fusion proteins they encode, remains poorly understood. Gene fusions commonly exert their oncogenic influence by either deregulating one of the involved genes (e.g. by fusing a strong promoter to a proto-oncogene), forming a fusion protein with oncogenic functionality (e.g. by causing a constitutive activation of a tyrosine kinase) or inducing a loss of function (e.g. by truncating a tumor suppressor gene)(qtd. in Latysheva and Babu 1).

One can attempt to infer a fusion protein's function by examining the structural or regulatory traits of its parents. A few studies do so for specific fusion proteins, trying to understand the underlying structural design principles of fusion proteins by studying the domain architectures of the parent proteins. But there will be lacunae in our understanding if we continue to study fusion proteins in such a manner.

After thoroughly researching fusion proteins we find that the root of understanding them lies in the sequence structure relationship. Since the fusion protein and its novel function stems from changes in the original sequence, the age old problem of protein folding is likely the focal point in understanding the how and why behind fusion proteins and their functions.

When studying fusion proteins several of our case studies were found to have an extra region in their sequence, one that matched neither of the two fused domains, and resided in between them acting as a linker. The linker is a highly disordered region and contains the translocation breakpoint, but we found very few studies on this region and no highly organized dataset of these regions exists. We found this very curious and decided to investigate further, on whether this could be a region of interest.

According to (Joseph et al. 1369-1381) inter-conversion between the *cis* and *trans* conformations has an important role in the folding process. Changes between *cis* and

trans conformations are found to be associated with the evolution of new functions facilitated by local structural changes. This is most frequent in enzymes where new catalytic activity emerges with local changes in the active site, which is very pertinent when looking at fusion proteins. Now how are changes in the cis and trans conformation facilitated? We believe the linker is responsible. It seems to be a vital structural feature when it comes to protein folding and conformational changes. An example of such a case has been studied by Sarkar et al. (413), the paper found that a proline on the linker tethering the two SH3 domains of the Crk adaptor protein interconverts between the cis and trans conformation. In the cis conformation, the two SH3 domains interact intramolecularly, thereby forming the basis of an autoinhibitory mechanism. Conversely, in the trans conformation Crk exists in an extended, uninhibited conformation that is marginally populated but serves to activate the protein upon ligand binding. This along with various other studies support the hypothesis that the linker in fusion proteins should be a region of interest.

In order to better understand the linker regions we studied two of their significant traits:

1. Intrinsic disorder.
2. Translocation breakpoint positions.

1.4. Significance of Disordered Regions in Protein Structures:

While it was thought for a long time that proteins fold into one defined tertiary structure numerous studies have since revealed that they in fact do not. Due to the function of many proteins, especially those involved in signalling pathways, intrinsic disorder is necessary for proteins to adapt to their environment with different conformations. These discoveries are now being used to establish the disorder–function paradigm , which states that certain polypeptide segments can be functional without achieving a defined tertiary structure (Babu 1185).

The structural plasticity and conformational adaptability of intrinsically disordered proteins/regions(IDPs/IDRs), their ability to react easily and quickly in response to

changes in their environment, and their binding promiscuity and unique capability to fold differently while interacting with different binding partners define a wide set of functional advantages of intrinsically disordered proteins over the ordered proteins(Uversky and Vladmir 1-6). These traits might be the reason why intrinsically disordered proteins play diverse roles in modulation and control of functions of their binding partners. Intrinsically disordered regions are very sensitive and dynamic regions, making them promiscuous binders which have the ability to form highly stable complexes and then once again adopt their highly flexible conformations after the completion of a particular function. They have an ability to adopt different conformations depending on the environment they are in and the function which is required by them. This is what makes them ideal in fusion proteins, where most functional fusion proteins have a disordered linker region (not aligning with either parent protein) in between the two domain structures.

Due to the crucial role played by intrinsically disordered proteins they are often regulated and tuned via alternative splicing and posttranslational modifications. Intrinsic disorder is necessary, however it is also a wildcard in that intrinsically disordered proteins can be implicated as causing multiple diseases such as cardiovascular disease, neurodegenerative diseases, diabetes, and our subject disease cancer. All of these factors support our claim that the intrinsically disordered linker region found in fusion proteins is a region of importance, and warrants further study.

1.5. Translocation Breakpoints in Intrinsically Disordered Regions:

Translocation breakpoints have been found to generally occur in intrinsically disordered regions, which may reflect a selection for regions that can more seamlessly combine different segments (Latysheva and Babu 4487–4503).The location of translocation breakpoints in cancer is known to be non-random and recurrent, and has been extensively demonstrated to be influenced by both the spatial proximity of chromosomes in the nucleus as well as features of the DNA sequence, such as repeats, fragile sites and endonuclease misrecognition sites (Latysheva and Babu 4487–4503). When observing

case studies before building our database we also observed that breakpoints preferentially avoid splitting domains. In instances where globular domains are split, it has been studied that the truncations tend to generate viable proteins due to the breakpoints being positioned in low hydrophobicity regions(Latysheva and Babu 4487–4503). This was also observed in our studies with fusion proteins, a few proteins seemed to have clean breaks without a disordered region between them. Looking at these cases we could say that the chimeric proteins are designed by nature to maintain viability and evade degradation pathways.

1.6. Proteasomal Cleavage Sites and Their Significance:

Proteins which are damaged or unneeded are generally degraded by proteasomes; These are enzymes which break the peptide bonds within the protein, rendering them non-functional. These proteasomes are responsible for cell cycle control, response to cellular stress, and also play important roles in the immune system. They have also been studied to increase the efficiency of degradation if large intrinsically disordered regions are present(Van Der Lee et al. 1832-1844). This is of particular interest in our study since the linker regions are highly disordered. We hypothesize that if we identify proteasomal cleavage sites present in the linker regions they could be targeted. My data could thus expose vulnerabilities that will enable future studies of potential therapies to inhibit oncoprotein function.

1.7. Recombinant Fusion Proteins:

As an indispensable component of recombinant fusion proteins, linkers at the fusion junction have great importance in the construction of fusion proteins as seen in the study by Amet et al. While in some applications the linker between the domains merely separates two protein domains and allows their independent folding, in many cases linker properties seem to be able to directly affect the functional properties of the fusion proteins in question.

The effect of linker insertion on expression level of fusion proteins was observed in a study by Amet et al. (523-528), Tf-fusion proteins designed for Tf receptor-mediated protein drug oral delivery in a study . Fusion proteins consisting of Tf and human growth hormone (hGH) were constructed in two directions (hGH-Tf and Tf-hGH) to test the optimal orientation; A helical (H4)2 linker (A(EAAAK)4ALEA(EAAAK)4A) was then inserted into the two fusion proteins (designated as hGH-(H4)2-Tf and Tf-(H4)2-hGH), greatly improving their expression level in transiently-transfected human embryonic kidney cells (HEK293) cells(Amet et al. 523-528). The hGH-(H4)2-Tf fusion proteins exhibited a 1.66-fold higher expression than hGH-Tf, while Tf-(H4)2-hGH had an expression level 2.39-fold higher than that of Tf-hGH(Amet et al. 523-528). This study also observed several other instances of linker insertion improving the expression level of fusion proteins. We believe that this is another point of interest, oncoproteins occur naturally and we believe that studying the insertion of a naturally formed linker in recombinant fusion proteins could be another area of research that could be explored once we establish a database. There is a possibility that studying the insertion of these naturally formed linkers could greatly improve the structure/function of recombinant fusion proteins, which leads credence to the fact that such a dataset is needed.

1.8. General Approach:

The basic idea is that by creating a database and storing all the linker regions from the available fusion proteins we should be able to analyze the large amount of data for different linker properties. Natural linkers adopt various conformations in secondary structure, such as helical, β-strand, coil/bend and turns, to exert their functions. Using this db we can find which conformation is preferred in these fusion protein linkers. Similarly we can identify hydrophobic, or hydrophilic properties, length, amino acid residue frequency, proteolytic sites, as well as the amount of disorder present in the linker regions.

There are several possible uses for this data, as described above they could be studied to improve recombinant fusion protein linkers. The linker lengths could be found, and we could research whether or not this is an important factor to understanding the change in protein function. If we find linker sequences which are unique they could help be used to further diagnostic studies.The sequence properties of these fusion linker regions could help us understand these structures engineered to cause cancer, along with the how and why behind their functions. And most importantly, as stated above the data could expose vulnerabilities that will enable future studies of potential therapies to inhibit oncoprotein function.

After understanding these studies and considering the possibilities we concluded that the linkers at the fusion junction are clearly significant to the functionality of oncoproteins and therefore warrant further research. However there's a roadblock of there not being any cohesive dataset of these regions. Therefore we endeavored to extract data and create a database consisting of oncoprotein linkers and information relevant to them.

Upon creation of the database, in hopes of unlocking more information we propose studying the proteolytic sites, surface accessibilities, and the lengths of these potentially unexploited vulnerabilities of cancer.

Chapter 2: Aims and Objectives

2.1. Vision:

To initiate a process where we create big data for fusion oncoproteins. Which can then be further explored by utilizing machine learning, deep learning, neural networks, and other AI tools for the purpose of enhancing diagnostics, improving targeted therapies, aiding future research, and contributing to research in more efficient design of recombinant fusion protein.

2.2. Aim:

There is a distinct lack of databases for fusion proteins, and an even more prevalent lack of ones with protein sequences. More importantly there is no database which has curated these fusion junction regions from fusion transcripts. Therefore there is a blind spot in our vision when it comes to data on fusion proteins; There is a *need* for us to build a cohesive dataset of these "linker" regions in order to be able properly visualize the problem and the solutions to it. We also endeavor to conduct preliminary investigation using machine learning(subject to time constraints).

2.2.1. Objectives:

1. Data collection:
    i. Obtain fusion protein sequences and create FASTA(fast-all) files.
    ii. To identify linker regions Basic Local Alignment Search Tool for proteins (BLASTp) will be used to perform searches against the non redundant protein database on the fusion protein sequences, and the results will be downloaded aseXtensible Markup Language (xml) files.
    iii. A dictionary to identify each corresponding gene in the fusion gene will be designed.
2. Appropriate libraries and python scripts will be used to further parse, clean the results, and isolate the linker regions.
3. A suitable database will be built to store the linker sequences and relevant data.
4. Use machine learning algorithms to observe trends in data.

2.3. Subsidiary Aim:

Obtain data in order to aid further research on these linker regions. Scour for proteolytic sites in the linker regions, further study their sequence and structural properties(Amino acid frequencies, amino acid propensities, secondary structures and lengths).

2.3.1. Objectives:

1. Data Collection:
    i. Proteases and their corresponding proteolytic sites will be obtained.
    ii. Python scripts will be used to identify proteolytic sites on the linkers obtained.
    iii. Other softwares will be located and used to find sequence and structural properties.
2. A few linkers will be treated as case studies to analyze the usability of the data.
3. Then the programs and softwares will be used on all the linkers detected and stored in the database. The results generated will be added to the existing database.

Chapter 3: Literature Review

The goal of this literature review is to give context and background to the study. A short summary of what exactly oncoproteins are, and how fusion oncoproteins are formed by chromosomal translocations. Then we discuss the current literature pertaining to cancer diagnosis and treatment, especially in relation to fusion proteins. From this we branch into the oncoprotein linkers/disordered regions at fusion junctions, and why they may hold interest according to some interesting studies. We conclude with a short discussion where we expose the blind spots in the research conducted so far, which lead to us designing the study conducted in this thesis.

3.1. Oncoproteins:

Proto-oncogenes are normal genes which are generally involved in cell division, cell differentiation, cell growth, and apoptosis. If a proto-oncogene undergoes some kind of genetic change they result in oncogenes. Oncogenes are so called due to their inherent role in oncogenesis, when they are activated they confer many different capabilities to a cancerous cell. Oncogenes are so sufficiently involved in tumor formation and maintenance that some tumors actually *require* oncogenic activation for prolonged sustenance.

We know of three basic methods which cause oncogene activation. These methods are mutations, gene amplification events, and chromosomal rearrangements (Pierotti et al. Section 6.2). Out of these mechanisms chromosomal rearrangements are of particular interest since they result in fusion proteins.

3.2. Chromosomal Rearrangement:

Chromosomal rearrangements are a feature often observed in hematological malignancies, as well as in some solid tumors. This makes them important features when researching treatments or diagnostics to several deadly cancers.

In the book Holland-Frei Cancer Medicine. 6th edition. Pierotti et al. , in (Section 6.2) it is stated that chromosomal rearrangements can lead to hematologic malignancy via two

different mechanisms: (1) the transcriptional activation of protooncogenes or (2) the creation of fusion genes.

In relevance to our topic we studied and summarized the excerpt on fusion genes from Pierotti et al. (Section 6.2). Fusion genes can be created by chromosomal rearrangements when the chromosomal breakpoints fall within the loci of two different genes. The resulting sequence consists of sequence segments from the parent genes, one gene being the head of the sequence and another gene being the tail. These chimeric proteins have transforming activity which is generally contributed to by both genes.

Our review of this section also found two examples of fusion genes which further prove their prevalence in deadly cancers, and explain how even though we know how they are formed we have little to no knowledge on their mechanism of action. There are large lacunae when it comes to our understanding of fusion proteins.

The first example of gene fusion was discovered through the cloning of the breakpoint of the Philadelphia chromosome in chronic myelogenous leukemia (CML); The t(9;22)(q34;q11) translocation in CML fuses the c-*abl* gene, normally located at 9q34, with the *bcr* gene at 22q11(Pierotti et al. Section 6.2). This fusion protein has increased tyrosine kinase activity and abnormal cellular localization. However although we know that the protein is of central importance in Chronic Myeloid Leukemia, and we have a certain understanding of the way it functions its mechanism of action is yet to be deciphered completely.

According to the study some solid tumors, like sarcomas, may have consistent chromosomal translocations that correlate with specific types of tumors. In sarcomas the majority of the fusion genes are involved in encoding transcription factors. In myxoid liposarcomas, the t(12;16)(q13;p11) fuses the *FUS* (*TLS*) gene at 16p11 with the *CHOP* gene at 12q13(Pierotti et al. Section 6.2). The FUS protein contains a transactivation domain that is contributed to the FUS/CHOP fusion protein, the CHOP protein is a dominant inhibitor of transcription and contributes a protein-binding domain and a presumptive DNA-binding domain to the fusion; Despite knowledge of these structural

features, the mechanism of action of the FUS/CHOP oncoprotein is not yet known(Pierotti et al. Section 6.2).

3.3. Cancer Diagnosis:

Cooper theorizes that the most effective method to treat the deadly disease that is cancer would be to prevent the initial development of the disease. If that's not possible then the second best alternative would be to try and detect it early, in the premalignant stages of tumor development. This way the treatment is easier, since the cancer hasn't metastasized yet localized radiation or surgery can be done to remove the cancerous cells.

For example, early stages of colon cancer usually require only a small surgical procedure, and are easily curable. However the cure rate for early carcinomas that remain localized to their site of origin is also high, about 90%; Survival rates drop to about 50% for patients whose cancers have spread to adjacent tissues and lymph nodes, and to less than 10% for patients with metastatic colon cancer(Cooper Section 15.5). Therefore early detection can be critical for a positive outcome when it comes to cancer treatment.

3.4. Cancer Treatment:

3.4.1. Pharmaceutical Therapies(Targeted vs. Non-Targeted):

Hait et al. examines details surrounding targeted and non-targeted therapies. Targeted therapy generally refers to a drug attacking, or rather targeting a specific molecule in order to inhibit its function. The drug is meant to selectively inhibit the target molecule which is functioning abnormally and leading to cancerous cells.

Non-targeted therapies on the other hand involve drugs discovered by phenotypic screening. These therapies are found without any prior knowledge of the target molecule, and tend to affect proteins or nucleic acids downstream of signaling pathways.

When the two therapies are compared several pros and cons can be seen. Whereas targeted therapies are highly effective in selected hematopoietic malignancies, most have shown limited efficacy against complex solid tumors(Hait et al. 1263-1267). In contrast,

nontargeted drugs are more successful and effective in general when compared to targeted therapies. However nontargeted drugs are also some of the most toxic drugs, unlike targeted therapies they tend to have several severe side effects which make them unpaletable to be used for treatment. Time costs are also associated with targeted therapies as molecular targets need to be identified before a drug can be designed, these time costs don't apply to nontargeted therapies.

3.4.2. Combination Therapies and Non-Pharmaceutical Routes of Treatment:

Cancer treatments that do not rely on the use of drugs are also prevalent. Nishida et al. states that surgery is the first option if possible to remove tumors, and toxic chemotherapy is generally a second resort when tumors cannot be removed surgically. However, combination therapy is thought to be the most successful method of cancer suppression; the success rate is reasonably higher when these cancer treatments are used in combination with drug therapies.

Combining chemotherapy and drug treatment is a popular alternative because it is less toxic than a complete chemotherapy regimen (Loeb et al. 776-781).

3.5. The Oncogene Addiction Phenomenon:

In a phenomenon known as oncogene addiction, cancerous cells rely on oncogenes and their products to play their specified role in order to function and survive (Jones and Thompson 537-548). For example, oncogenic addiction may force cancer cells to rely on a certain metabolic pathway for growth (Jones and Thompson 537-548). There is a saying that needs to be heeded without fail during any war, "An army marches on its stomach" . Similarly the cancer cells declare war on our body, however they can only grow and metastasize if they are adequately supplied. Boroughs and Deberardinis say in their paper that the activation of oncogenes and loss of tumour suppressors causes metabolic reprogramming to take place in the cell, leading to increased nutrient uptake to supply the cancerous growth.

A good example to observe would be Ras-related oncogenes; Oncogenic Ras stimulates both glucose uptake via enhanced expression of GLUT1, and utilization of glucose by

anabolic pathways. Ras also regulates glutamine metabolism, specifically directing glutamine carbon into pathways that support biosynthesis, redox homeostasis and ultimately cell survival and growth(Boroughs and Deberardinis 351–359).

This phenomenon of "Oncogene addictions" means that there is a dependency on one or a few genes in some cancers in order to maintain their malignant phenotype. This dependency means a weakness for cancer, which could be exploited to cut off their supply. Weinstein et al. found that the most convincing evidence for the concept of oncogene addiction comes from the increasing number of examples seen of antibodies or drugs that target specific oncogenes showing great therapeutic efficacy in the treatment of cancer.

A great example for oncogene addiction and how it could be considered an achilles heel for cancer can be observed in the case of the proto-oncogene HER2/neu. This oncogene is present as a proto-oncogene in normal cells, however it becomes an oncogene when it is produced in abundance leading to a large excess of the protein being present in cancerous cells. Gutierrez and Schiff have observed that cancerous cells that have a high amount of HER2/neu protein didn't respond as well to chemotherapy when compared with patients who didn't have the HER2/neu protein in large excess in their cells. These studies have led to drugs being made available to target the HER2/neu protein in order to improve the prognosis of patients, and great success has been observed with drugs such as Herceptin®. Baselga (14-21) states in a study that Herceptin in combination with chemotherapy, in particular paclitaxel, significantly improved time to disease progression, duration of response and time to treatment failure. Combination therapy was also seen to provide a significant 25% improvement in survival rates.

Weinstein et al. proposed that the phenomenon of oncogene addiction is a consequence of the fact that the multistage process of carcinogenesis is not simply a summation of the individual effects of activation of multiple oncogenes and inactivation of multiple tumor

suppressor genes. He hypothesizes that because these proteins tend to have multiple roles in complex and interacting networks, the intracellular circuitry which regulates signal transduction and gene expression in cancer cells is very bizarre in comparison with that of normal cells. Therefore an oncogene could play a very imperative and different role in cancer cells' given pathways than it does in normal cells.

3.6. Disordered Regions at Fusion Junctions, How They are Interesting:

Intrinsically disordered proteins have often been a common factor for various diseases such as various neurodegenerative and cardiovascular diseases, as well as cancer and diabetes. This has made them the subject of intense research in recent years. Uversky and Vladmir state that these proteins seem to possess a certain conformational adaptability, with intrinsically disordered regions displaying a structural plasticity that may lend to them having the ability to react more readily in response to changes in their environment. In addition to this, their binding promiscuity, their ability to fold differently when interacting with varying binding partners, sets them up to have certain functional advantages over normal proteins. These advantages allow the proteins to play more diverse parts in processes, such as playing roles in cell signalling; Abundant involvement of intrinsically disordered proteins has been observed in processes such as cell regulation, or recognition as well. The intrinsically disordered proteins also, due to their inherent traits, are able to modulate and to some extent control the functions of the proteins they bind to. When they do bind to proteins they are known to form highly stable complexes; in signaling interactions they have been observed to undergo bound and unbound transitions, acting like very dynamic and sensitive on-off switches.

Gsponer et al. alludes that since intrinsically disordered proteins are very common and possess various crucial functions, as well as playing important roles in several vital biological processes, they must be highly regulated.In most processes it is crucial that the protein should be available in certain appropriate amounts, and not be present for long periods of time when not needed. This leads to the hypothesis that intrinsically disordered proteins are very strictly controlled and regulated by processes in order to not

malfunction. To check this hypothesis, careful analysis of the IDP regulation inside the cell at different stages of protein synthesis and degradation was recently conducted using the corresponding data available for the *Saccharomyces cerevisiae, Schizosaccharomyces pombe*, and *Homo sapiens* proteomes (Gsponer et al.1365–1368). This analysis revealed that IDPs were less abundant than ordered proteins in these tree proteomes because of the increased decay rates of mRNAs encoding IDPs, lower rates of IDP protein synthesis, and shorter half-lives of IDPs (Gsponer et al.1365–1368). Also, the majority of IDP-targeting kinases were either regulated in a cell-cycle dependent manner, or were activated upon exposure to specific stimuli or stress (Gsponer et al.1365–1368), this added additional control over the disordered proteins. It is important to remember though that although the abundance of many IDPs is tightly regulated, some disordered, and hybrid proteins are present in cells in large amounts, and for long periods of time due to either specific post translational modifications, or via interactions with other factors, or due to their localization in specific compartments (Uversky and Vladimir 1-6).This might aid some disordered proteins in evading degradation.

It's obvious that when a process is tightly controlled it is because loss of control could lead to dire consequences. These consequences have been observed in the development of pathological conditions in several human diseases with the loss of regulation and control over intrinsically disordered proteins. Uversky observed that the failure of such a protein to adopt, possess, or keep functional state is commonly associated with protein misfolding, loss of normal function, gain of toxic function, and/or protein aggregation. It has also been studied that some disease-related proteins form pathogenic conformatons because they have an intrinsic propensity for it. For other proteins, interactions or impaired interactions with chaperones, intracellular or extracellular matrices, other proteins, small molecules, and other endogenous factors can induce conformational changes and increase propensity to misfold; All these factors can act independently, additively, or synergistically (Uversky 4191 - 4213).

Uversky and Vladimir state that one of the most radical and obvious ways to generate a pathological protein is chromosomal translocation, which generates chimeric proteins by fusing segments of two otherwise separated genes. Several forms of cancer are thought to be directly, or indirectly caused by chromosomal translocations.

In a study conducted by Hegyi et al. Computational analysis of 406 chimeric proteins showed that oncoproteins are imbued with several disordered regions, and translocation breakpoints are normally located in a region which shows almost two times the disorder that the already highly disordered protein has. The proteins show a pattern where the functional domain normally doesn't contain the translocation breakpoint and is an ordered region, while the rest of the protein is significantly disordered, more so at the site of the translocation breakpoint. The study also implies that if the breakpoint falls in a domain then the truncated domain being fused would lead to a folding-incompetent protein. Such a protein would be degraded quickly, and not be able to add any value to tumorigenesis.

Hegyi et al. thus noted that a functional protein would fuse in a disordered region, because intrinsically disordered regions don't have a lot of hydrophobic amino acids, similar to linker regions. Any misfolded protein is generally a target to be degraded in the cell, the major identifier of misfolded proteins is thought to be solvated hydrophobic amino acids. If such amino acids are present cellular-destruction mechanisms such as the ubiquitin–proteasome pathway will likely be activated to destroy the misfolded protein. Linker sequences, due to their function, are unfolded and also lack hydrophobic residues, thus disordered linker regions are not classified as misfolded, and a fusion taking place in such a region does not cause major structural disturbances(Uversky 343–384).

In addition to allowing the protein to survive structural disorder may also be a factor that allows the novel functions gained by fusion proteins.Long intrinsically disordered regions may lend the flexibility required to allow interaction of the functional domains brought

together by fusion. An interesting example of this can be observed by a study conducted on the fusion protein EML4-ALK where the influence of the disorder region with the breakpoint was further researched by Wang et al. The EML4-ALK variant 1 fusion protein has functional domains from both of the parent proteins. The overall protein is two domain structures connected by a disordered linker region. As the normal function of ALK protein needs the dimerization triggered by signals from the extracellular domain, the fusion of EML4-ALK variant 1 containing a dimerization motif from EML4 might contribute to the dimerization of the EML4-ALK variant 1 and thereby trigger the autophosphorylation of the kinase domain and lead to the oncogenic potential in non-small-cell lung cancer (Mano 193-201). This logically leads us to believe that the disordered linker region in between the two domains is facilitating the interaction between them, and thus allowing the protein's oncogenic activity.

3.8 Targeting Linker Regions:

To design treatments targeting intrinsically disordered regions is difficult, due to how flexible and dynamic the regions inherently are. Seeing that these regions are extremely prevalent in several human diseases however means that it is imperative some research be conducted on them. In the past year different experimental and computational approaches, as well as the combination of both, have been explored to identify molecules to target either the hot-spots or the allosteric sites of IDPs(Santofimia-Castaño et al. 1695-1707). One of the most successful approaches according to the research conducted by Santofimia-Castaño is using small molecules. The study states that in all cases of targeting IDPs so far the molecules interfere with DNA–protein or protein–protein interactions which require the intrinsically disordered region.

Two perspectives exist when it comes to using small molecules to target. The first perspective involves identifying some short intrinsically disordered region that may be imperative in protein function. These molecules will replace the recognition region of the IDP in the binding to its partners (Cheng et al. 434) The second perspective is achieved when a small molecule, identified by some screening protocol, binds to an important

intrinsically disordered region, usually, but not always at a hot spot. The hot spot being a region which is vital for the protein; Any hindrance of this disordered region leads to the protein being unable to function.

According to Santofimia-Castaño et al. the designed compounds against IDPs encompass small synthetic molecules, or peptides. This is very similar to the designed compounds against ordered proteins. The difference lies in the design, in well folded proteins the design of molecules to act against them is done in a structure based fashion. The case is different when it comes to intrinsically disordered proteins, where due to lack of structure, the design is essentially ligand based. We don't have a single defined methodology established against IDPs yet, and our ability to design drugs targeting IDPs remains stunted due to lack of understanding of the sequence and structure relationship. However, as John Maynard Keynes famously said, "The difficulty lies, not in the new ideas, but in escaping from the old ones."

3.9 A Basic Introduction to Proteases and Their Significance:

Proteases are an intrinsic feature in prokaryotic and eukaryotic organisms. When proteins are rendered defective due to misfolding or damaged in any way, broad-specificity intracellular proteases tend to degrade them. When these proteins play important roles such as regulating the cell cycle, selective degradation is an imperative process. For proteases to play their roles efficiently and successfully cleave damaged proteins, while also avoiding off-target damage they must be well regulated. This regulation generally takes place by sequestration in a separate compartment, that is, a subcellular organelle or a protein chamber, and by controlled substrate delivery(Suskiewicz et al.1285-1297). Often compartmentalization will enhance and bring about the most efficient proteolytic degradation since several different proteases are involved in degrading the protein together.

A good example to visualize compartmentalization better would be the organelle lysosome.Autophagy is an intracellular process which utilizes lysosomes in order to

degrade proteins. Lysosomes are membrane-enclosed sacs which contain various proteases and digestive enzymes. Since the proteases are contained within the lysosome they are unable to damage other proteins, and only take effect on proteins which are taken up by the lysosome. Autophagy does this by forming autophagosomes from membranes derived by the ER. These membranes enclose small areas of cytoplasm and/or cytoplasmic organelles, these vesicles then fuse with lysosomes, and the degradative lysosomal enzymes digest their contents(Cooper Section 7.4).
The lysosomal pathway, in comparison to the proteasomal pathway which is another method to regulate proteasomal degradation, is typically non-selective.

Proteasomes are protein complexes which lead to the degradation of damaged or misfolded protein via proteases. These are involved in selective degradation, based on the presence of ubiquitin. This ubiquitin sequence is a 76-amino-acid polypeptide, and acts as a modification to proteins which are to be targeted for degradation.Additional ubiquitins are added over time to form a multiubiquitin chain. The proteasome, upon recognizing this ubiquitin marker, uses proteases to rapidly degrade the protein into its amino acids.These amino acids can then be used for the synthesis of another protein. Most vital processes, such as gene expression and cell proliferation, involve proteins that need to be highly regulated by ubiquitination and proteolysis.

But degradation isn't the only important trait of proteases. There have been several cases observed where activation of a protein takes place when its cleaved with a protease.The sequence cleaved off the precursor protein is called a propeptide, and the protein which remains is termed the proprotein. One example of a protein derived from a proprotein would be insulin. In mammals preproinsulin is not active as an hormone when it is first translated from the mRNA. It becomes active by this process: First the N terminal sequence is cleaved off , then it forms internal disulfide bonds, and then a sequence called C-peptide is cleaved out of the proinsulin. The two end pieces are still connected by internal disulfide bonds, leading to active insulin hormone.

Another interesting case of proteolytic activation is described in a study by Andréasson et al. The SPS-sensing pathway exists in eukaryotic cells and requires proteolytic activity in order to function. It utilizes direct activation of a protease to mobilize latent transcription factors by proteolytic processing. The transcription factor Stp1 is endo proteolytically processed in response to extracellular amino acids by the plasma membrane SPS (Ssy1–Ptr3–Ssy5)-sensor. Processed Stp1, lacking a cytoplasmic retention motif, enters the nucleus and induces amino acid transporter gene expression. The SPS-sensor component Ssy5 is a chymotrypsin-like protease with a Pro-domain and a catalytic domain. The Pro-domain, required for protease maturation, is autolytically cleaved from the catalytic domain but remains associated, forming an inactive protease complex that binds Stp1. Stp1 is processed only after amino acid-induced signals cause the dissociation of the inhibitory Pro-domain. These findings by Andréasson et al. demonstrate that gene expression can be controlled by regulating the enzymatic activity of an intracellular endoprotease.

Hubbard et al. suggested that for a protein to be cleaved by a protease, it must adopt an extended conformation that requires a propensity for local unfolding of at least 12 residues around the scissile bond. This was proved in the study when it was subsequently seen that subtilisin cleavage sites in thermolysin correspond to peaks in X-ray-derived B-factor values. These peaks would be higher for regions without secondary structure, such as flexible linker regions. It was later demonstrated in the same study that regions without electron density in the cleavage structure overlap with cleavage site regions, leading to the belief that cleavage sites are predominantly present in intrinsically disordered flexible regions lacking secondary structure. This is in congruence with the observation that one of the features commonly associated with IDPs is their susceptibility to proteolysis(Hubbard et al. 349-359).

3.10 Gaps in Literature:

After understanding these studies and considering the possibilities we concluded that the linkers at the fusion junction are clearly significant to the functionality of oncoproteins and therefore warrant further research. From our literature survey we can identify that there's lack of a large dataset of these linkers, and there has also not been a large organized study conducted on them as of now. Therefore we endeavored to design a study that might be a missing puzzle piece in the overall image of fusion oncoproteins. We believe that the disordered “linker’ regions containing the translocation breakpoints are of importance to the proteins’ structure and activity. By identifying these linkers and creating a large database we will be capable of unlocking informative trends and traits that could be crucial to improved targeted therapies to cure cancer.

Chapter 4: Materials & Methods

4.1 Methods:

4.1.1. Terrain Mapping:

We found the protein sequences of about 20 fusion proteins from various sources such as the Uniprot and FusionGDB databases, as well as the Catalogue Of Somatic Mutations In Cancer(COSMIC) database. We then used blastp to BLAST these sequences against the non-redundant protein database, filtering for Humans(taxid:9606). Since the rest of the settings remained default we ended up with 100 alignment results. We noted that in a few case studies we only got alignments against one of the parent proteins when the result limit was set as 100 and when we increased this limit we got alignments with both parent proteins. In the proteins where we did get alignments with both parent proteins we observed two cases in terms of linkers:

1. There was no linker region, and the proteins were perfectly fused together.
2. There was a gap between the alignments of the two parent proteins, meaning a linker region was present.

These cases can be seen in Figure 3. With colored regions belonging to parent proteins.

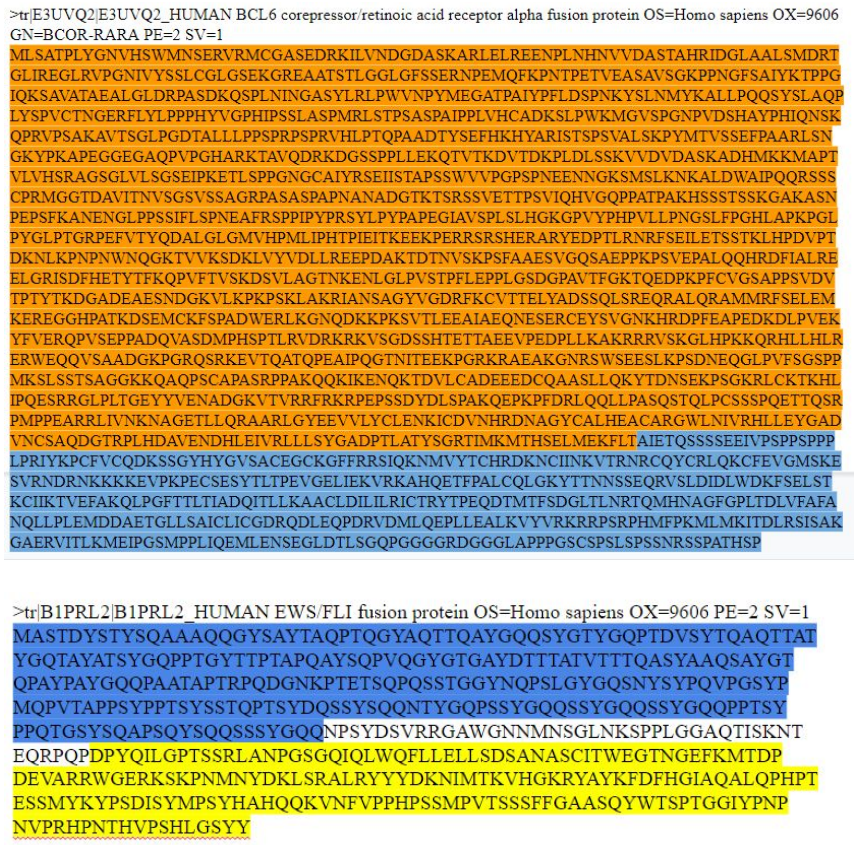


Figure 3. A visual representation of the two cases we observed in our pilot study.

We observed these regions and found the linkers from our case studies were all in highly disordered mobile regions or coil regions according to the disordered region prediction server Disembl. Then we checked their amino acid properties, and how they matched up to definitions of traditional linkers in normal non-oncogenic proteins. Using the ProtScale tool we found that the regions largely avoided hydrophobic residues, and displayed good flexibility, similar to other protein linker regions(Gasteiger et al. 571-607).

Now that we had determined a method to extract the linkers, and what our results should be, we decided to automate the process in order to obtain a larger dataset. Since the more data we have the better we can gauge and correlate the linker properties.

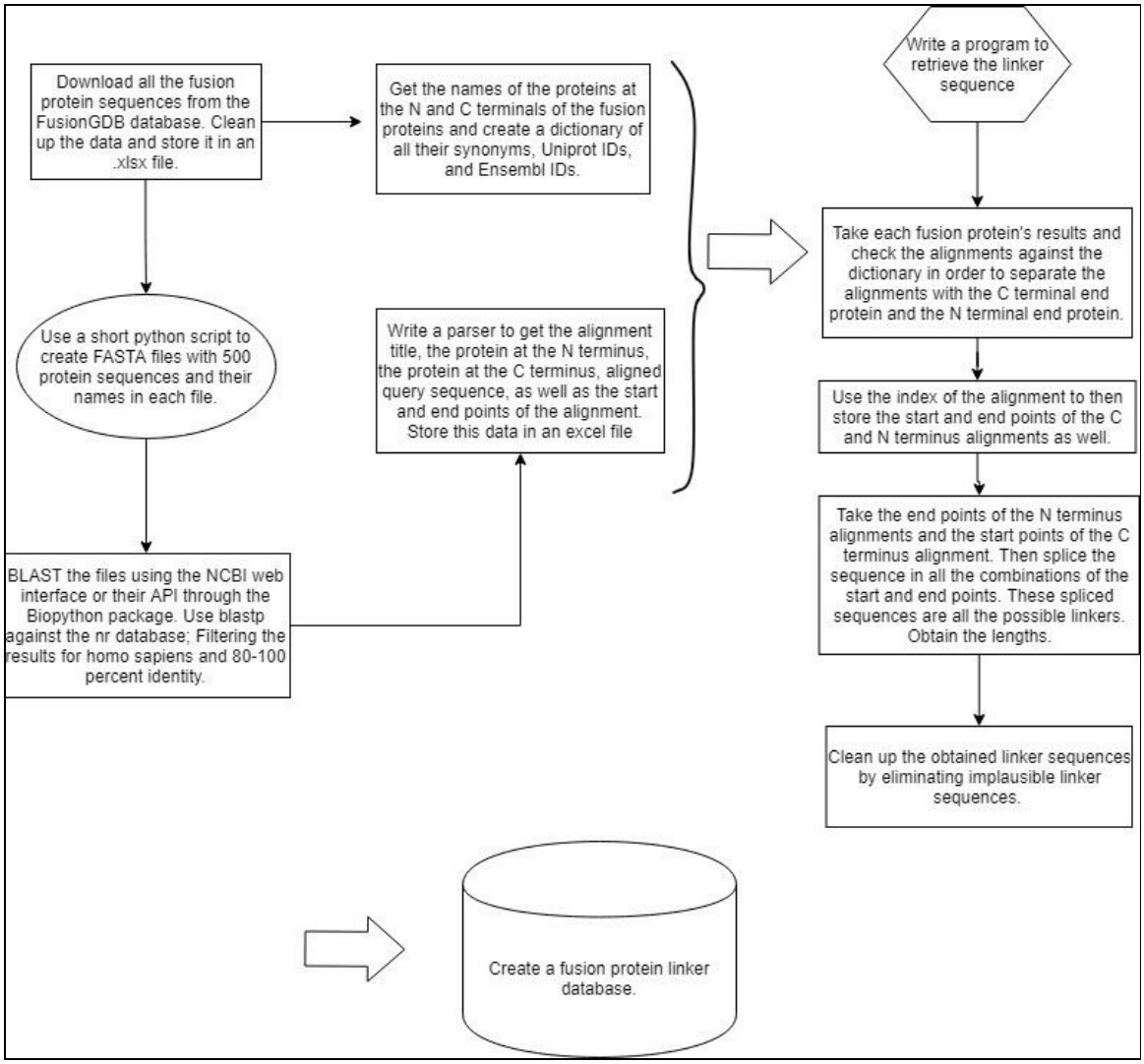


Figure 4. A Flowchart demonstrating the process utilized in order to achieve creation of the Fusion protein linker database. This representation is meant to give a brief overview, with the sections being described in more detail in the body of the text.

4.1.2.Data Collection

4.1.2.1. Origin of Protein Sequences:

We downloaded the dataset of fusion protein sequences from FusionGDB, since it seemed to have the most exhaustive collection of protein sequences when compared with other databases such as COSMIC. The FusionGDB database has 48,117 fusion genes from three well known databases: the improved database of chimeric transcripts and

RNA-seq data (ChiTaRS 3.1), an integrative resource for cancer-associated transcript fusions (TumorFusions), and The Cancer Genome Atlas (TCGA) fusions by Gao et al. In addition to this they have translated amino acid sequences, which are preferable to the DNA transcripts featured in most other databases. Therefore as a starting point we downloaded the protein sequences from FusionGDB and decided we would expand our dataset to include data from other databases at a later stage.

At the time of download we obtained 57,613 protein sequences in a txt file(Due to Isoforms and varying nucleotide transcripts). The information included was as such: ORF-Henst-Tenst-empty-DataType-Source-Ctype-Sample-Hgen-Hchr-Hbp-Hstrand-Tge ne-Tchr-Tbp-Tstrand-SeqLen-Seq. Since we wanted to create FASTA files of these sequences so that we can BLAST them our columns of interest were the Head gene, the Tail gene, and the Sequences. Therefore upon downloading the data we opened it with notepad and used word wrap on it. Following this we opened the file as an xlsx file and split the text into columns by the space as a seperator. Then, as seen in the code below, we parsed the xlsx file in order to generate FASTA files containing batches of 500 fusion protein sequences each. The reasoning behind there being only 500 sequences per file will be further explained in the BLAST section.

```
In [4]: import xlrd
        import os.path
        os.chdir(r"C:\Users\svpan\Downloads\All_seqs")
        wb = xlrd.open_workbook('all fusions.xlsx')
        wb.sheet_names()
        sh = wb.sheet_by_index(0)
        a=0
        list1=[]
        while a<=57000:
            list1.append(a)
            a+=500
        for jk in list1:
            i = 0
            jk2=jk+500
            name1="seqs_"+str(jk2)+".fasta"

            print(jk)
            print(jk2)
            with open(name1, "a") as my_file:
                for i in range(jk,jk2):
                    sequence = sh.cell(i,17).value
                    nterminus = sh.cell(i,8).value
                    cterminus = sh.cell(i,12).value
                    if sequence:
                        DB1 = (">"+ nterminus + "/" + cterminus + "\n" + sequence)
                        my_file.write(DB1 + '\n')
                    i = i+1
```

Figure 5. The short python script used to create fasta files containing sequences and fusion protein names.

4.1.2.2. Sequence Alignment:

In order to gather more linker regions a fundamental step was to do local alignment of the fusion protein sequence with the database of non-redundant protein sequences, in order to obtain alignments with both parent proteins. Once these alignments were removed from the query sequence we would end up with the linker region. However in order to do this alignment we had two possible algorithms available. We could download the nr database and try to run the Smith-Waterman algorithm with it, or we could use BLAST(Basic Local Alignment Tool). BLAST allows the user to compare a protein sequence with a database of protein sequences, and identifies sequences that resemble the query sequence above a certain threshold. Similarly the Smith Waterman algorithm also performs local

alignments the difference being that it is far more exhaustive when compared with BLAST(Schpaer et al. 179-191).

Shpaer et al. states that while both Smith-Waterman and BLAST are used to find homologous sequences by searching and comparing a query sequence with those in the databases, they do have their differences.The BLAST algorithm is a development of the Smith-Waterman algorithm suggesting a time-optimized model contrary to the more accurate but time consuming calculations of the Smith-Waterman algorithm (Altschul et al. 403-410). Due to this fact, the results received through BLAST, in terms of the hits found, may not be the best possible results, as it will not provide you with all the hits within the database. A better alternative in order to find the best possible results would be to use the Smith-Waterman algorithm, which is better in accuracy and speed. The Smith-Waterman algorithm doesn't miss any matches, unlike BLAST. However it requires large amounts of computer usage and space.

After weighing the pros and cons we decided that using BLAST was the best option. Since we aren't looking for sequences with distant homology BLAST, due to its speed and ease of use, sufficiently met our requirements.

4.1.2.3. BLAST:

There are four options available when you want to BLAST large numbers of sequences:

1. BLAST API:. The NCBI servers are a shared resource and not intended for projects that involve a large number of BLAST searches. NCBI provides Stand-alone BLAST and the RESTful API at a cloud provider for such projects. Therefore the API wasn't a great option for us to use. However we tried it anyway, unsuccessfully, through the NCBIWWW package provided in BioPython. When we programmatically submitted BLAST searches using the API via python we faced difficulties such as being unable to filter by taxon:9606, unreliable internet connection leading to incomplete xml result files being downloaded, and unreliable search times(sometimes the results would be returned quickly and all the files would be perfectly fine; other times it could take upto three hours or overnight for the result files to be generated and several of them would be incomplete xml files).We believe these issues occurred due to our internet connection being

unreliable, the BLAST servers being overloaded, race conditions, and other server side issues in addition to the result file size being pretty large. The API option was also rejected by us due to BLAST stating that they will move searches of users who submit more than 100 searches in a 24 hour period to a slower queue, or, in extreme cases, will block the requests; This may have also been the reason why we sometimes had to wait overnight for searches to complete. Therefore due to the aforementioned reasons, and since we had a time constraint this method wasn't very feasible.

```
In [3]: from Bio.Blast import NCBIWWW

        sequence_data = open("seqs_2000.fasta").read()
        sequence_data
        result_handle = NCBIWWW.qblast("blastp", "nr", sequence_data, hitlist_size=500)

        with open('results.xml', 'w') as save_file:
            blast_results = result_handle.read()
            save_file.write(blast_results)
```

Figure 6. The trial code which was used to test the BLAST API option.

2. Stand Alone BLAST: This is a version of BLAST which can be installed on your system and used locally. We installed Stand Alone BLAST, and the non redundant protein database in order to see if this was an option for us to use for our project. Then we held several trial runs for over three months with standalone BLAST, where we tried to get better RAM for the system we were working on, changed our OS to UNIX, and tried to get a better processor. Eventually we managed to get a system with UNIX, 16GB RAM, and an i5 processor. The system requirements were not good enough for us to be able to run BLAST; Especially due to the large amount of files we had, it would take a day or more for one file to be processed. Waiting any longer for a better system after already being halfway through the time provided for the thesis project was not a great idea, therefore we had to abandon this option as well.

3. AWS BLAST: NCBI provides a BLAST server image at Amazon Web Services (AWS).This was the best option for our project, since it provides a scalable virtual memory where we can increase the memory and processing power depending on our needs. The rate is also very practical when compared with having to actually buy the hardware and scale up our own systems. But due to lack of funding we had to abandon this option as well.

4. Web BLAST: This involves using the web interface provided by NCBI. Using this interface requires me to manually run the searches for each file. However with the other options exhausted we had no option but to take this route, which ended up not being the best. We could only set a limit for 100 alignments since otherwise the file was too big and due to faulty internet we would end up with partial xml files almost always. In addition to this BLAST would discard my searches. It was rarer for the download to fail when the limit was 100 alignments, but it *did* still happen. This led us to having to redo several results files since the partial xml file was only discovered during program runtime.

This bottleneck set us back quite a bit, especially since only 100 alignments wasn't suitable for all fusion proteins since both parent proteins didn't always show up. But since we were out of options we resigned ourselves to go back and redo the BLAST searches for those proteins which didn't generate results with a higher upper limit for alignment results. Due to us having fewer alignments we were guaranteed to have fewer protein linkers show up if we filtered them too stringently. Therefore we relaxed the constraints and only filtered for 0.5 e-value in our pilot studies; We deemed this sufficient, especially since there were several transcripts for the same proteins in the results. In addition to this we only took the first, which is the best scoring, High-scoring Segment Pair (HSP). HSP is a local alignment with no gaps that achieves one of the highest alignment scores in a given search; The extent to which two (nucleotide or amino acid) sequences have the same residues at the same positions in an alignment, often expressed as a percentage(Altschul et al. 403-410). In addition to this according to Altschul et al the lower the e-value, or the closer it is to zero, the more "significant" the match is. However, virtually identical short alignments have relatively high e-values. This is because the calculation of the e-value takes into account the length of the query sequence. These high e-values make sense because shorter sequences have a higher probability of occurring in the database purely by chance(Altschul et al. 403-410).

4.1.2.4. Parsing the BLAST Results to Obtain Linker Regions:

We chose to download the outputs in xml format since it contains markup symbols to describe the contents of the data within the file. It is also the only output file which contains all the details of the BLAST alignments such as definition line, the start and end points where the query sequence aligned with the alignment, and all the scores necessary to filter the BLAST records.We proceeded to use the BLAST xml parser available in the BioPython package to parse the results files, adding our own code where necessary. The result files were parsed to extract the start point of the alignment with the query sequence, the end point of the alignment with the query sequence, and the hit definition of the alignment; All while filtering out any alignments which didn't have the highest HSP. The reason we took the hit definition rather than the hit accession was due to the large variety of databases within the non redundant protein database. We would have had a more difficult time separating the accession IDs and searching for them in their respective databases than we would simply using the hit definition line. We stored these three parsed items along with the protein at the C terminal, N terminal, and the sequence in a dataframe. This dataframe was then written into an excel file.

Concurrently we were building a dictionary of fusion genes so that the hit definition line could be parsed as accurately as possible. First we took all the N and C terminal proteins in the FusionGDB database file and used the Retrieve/ID mapping tool available on UniProt; We searched for their UniProtKB identifiers. We then downloaded the Ensembl identifiers, UniProt IDs, Gene synonyms, Gene names, Protein names, and the Genes(Entry) for all of the proteins. However the Retrieve/ID mapping tool didn't find all the proteins. Therefore following this we wrote a short python script that would check the N and C terminal proteins in the excel file containing the parsed results for any missing genes which were not in the dictionary. Once these missing genes were all identified we manually entered information on the 35 or so genes by finding them on Genecards or Ensembl.

4.1.2.5. Extracting the Linkers:

The principle behind extracting the linkers was fairly simple. We simply wanted to separate all the alignments with the N terminal protein and all of the alignments of the C terminal protein for each respective fusion protein. Then we would slice the fusion protein sequence at the end point of the fusion protein's alignment with its N terminal protein, and at the start point of the fusion protein's alignments with its C terminal protein. However the program was reasonably more complex than that. The process flow for the program can be seen in Figure.In a short simplified summary: the program consisted of two major functions.

The first function used the gene dictionary to filter for the N and C terminal protein alignments for each protein by searching alignments' hit definition strings. It first obtained the name of the fusion protein, and split it to get the names of the parent proteins. It then scoured the string for any mention of the terms in the gene dictionary corresponding to the name of the fusion protein's N terminal end protein or C terminal end protein. It then separated them and stored each alignment's start and end point, as well as the fusion protein name and sequence in a dataframe.

This data was then passed into the second function which would filter out the alignments for which the start point for the C terminal protein or the end point for the N terminal protein did not exist, this means that no acceptable alignments were found for the N or C terminal proteins. In this case "No linker found" would be appended to the array which would later be put into a dataframe. Following this the program further filtered for any errors such as alignment where the start point for the C terminal protein or the end point for the N terminal protein were equal to each other, or if they overlapped. These alignments would also be discarded. Following this depending on whether more alignments were found for the N terminal protein, or the C terminal protein nested loops were used to slice the sequence at the end points for the N terminal protein and start

points for the C terminal protein and put all the linkers into an array which would later be put into a dataframe. Within this loop all the end points for the N terminal protein and start points for the C terminal protein were also subtracted to gain the linker lengths. These linkers and their corresponding lengths were then once again put into a dataframe containing the protein name, protein sequence, the end point for the N terminal protein, the start point for the C terminal protein, as well as the linker sequence and its length.

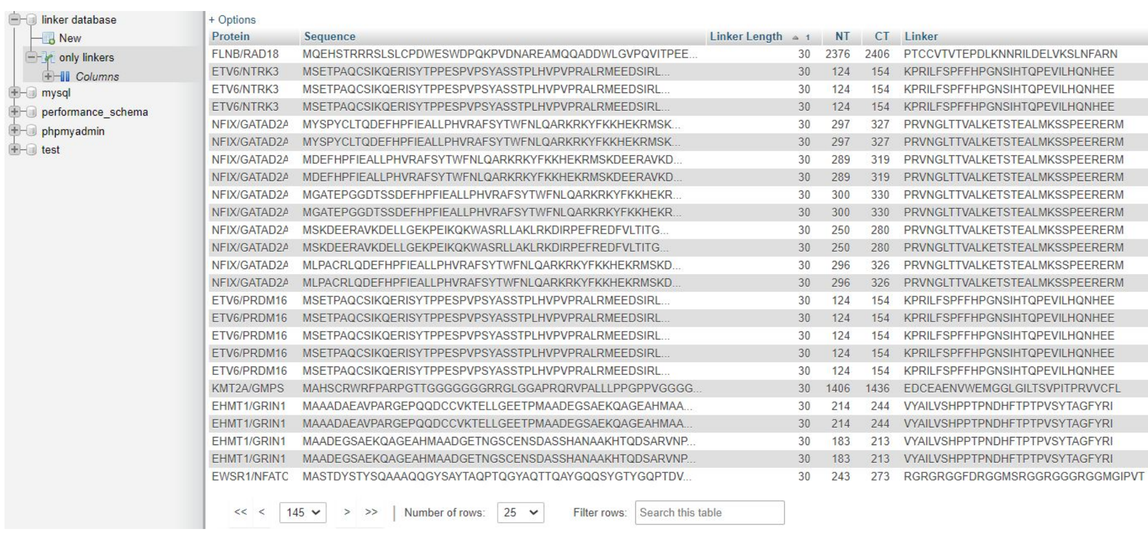
The program then generated two files. First an intermediary file which contained each alignment's start and end point, as well as the fusion protein name and sequence. Second the actual result file which contained the protein name, protein sequence, the end point for the N terminal protein, the start point for the C terminal protein, as well as the linker sequence and its length.

We ended up with a pretty decent amount of linkers. However we hadn't cleaned or quality checked the data yet. After doing a few manual checks by going through a certain result file and finding the linkers myself in order to compare my results with the database we ended up finding several duplicate results due to there being multiple alignments of the same protein for the fusion proteins. In addition to this we realized that the large number of files were a bit of nuisance to deal with and decided we wanted to concatenate them for ease of use. But due to the large amount of data we realized it was not feasible to put all of it into one excel file. Therefore we split up the data and concatenated it in 4 batches, and generated 4 excel files containing all the linker data. A minor correction was made to the file so that the protein name was printed on every line until the next protein record, since the previous files only printed the fusion protein name at the beginning of each record. The resulting files had 56.6 MB of data. Following this correction the files were run through a program to remove the duplicate results. This was done by concatenating the elements of each row into a string with delimiters in between each element and storing it in an array. This array was then converted into dictionary keys and then back to an array. Since dictionary keys must be unique this automatically removes

any duplicates. Following this the string was cut at the delimiter and the elements were once again stored into a dataframe which was converted into an excel file. These excel files were then concatenated into one file which had 4.62 MB of data after the cleaning.

4.1.2.6. Creating the Database:

We converted the cleaned excel file into a csv file using a short python script and stored the data into a single table in MariaDB using the Cross platform - Apache, MySQL, Perl and PHP (XAMPP) interface. Following this we decided, for ease of use, to create a user interface using Hypertext Markup Language(HTML) and Structured Query Language(PHP) to interact with the database and provide the possible linker regions in a table format for any fusion protein a user enters. The user base in mind when designing this database was simply the researchers in the lab where the work for this dissertation was carried out. We hope to scale up the design and functionality of this interface in the future, as we expand and improve upon this pilot study.



Protein	Sequence	Linker Length	NT	CT	Linker
FLNB/RAD18	MQEHSTRRRSLSLCPDWESWDPQKPVDNAREAMQQADDWLGVPQVITPEE...	30	2376	2406	PTCCVTVTEPDLKNNRILDELVKSLNFARN
ETV6/NTRK3	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
ETV6/NTRK3	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
ETV6/NTRK3	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
NFIX/GATAD2A	MYSPYCLTQDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKRMSK...	30	297	327	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MYSPYCLTQDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKRMSK...	30	297	327	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKRMSKDEERAVKD...	30	289	319	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKRMSKDEERAVKD...	30	289	319	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MGATEPGGDTSSDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKR...	30	300	330	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MGATEPGGDTSSDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKR...	30	300	330	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MSKDEERAVKDELLGEKPEIKQKWASRLLAKLRKDIRPEFREDFVLTITG...	30	250	280	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MSKDEERAVKDELLGEKPEIKQKWASRLLAKLRKDIRPEFREDFVLTITG...	30	250	280	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MLPACRLQDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKRMSKD...	30	296	326	PRVNGLTTVALKETSTEALMKSSPEERERM
NFIX/GATAD2A	MLPACRLQDEFHPFIEALLPHVRAFSYTWFNLQARKRKYFKKHEKRMSKD...	30	296	326	PRVNGLTTVALKETSTEALMKSSPEERERM
ETV6/PRDM16	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
ETV6/PRDM16	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
ETV6/PRDM16	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
ETV6/PRDM16	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
ETV6/PRDM16	MSETPAQCSIKQERISYTPPESPVPSYASSTPLHVPVPRALRMEEDSIRL...	30	124	154	KPRILFSPFFHPGNSIHTQPEVILHQNHEE
KMT2A/GMPS	MAHSCRWRFPARPGTTGGGGGGGRRGLGGAPRQRVPALLLPPGPPVGGGG...	30	1406	1436	EDCEAENVWEMGGLGILTSVPITPRVVCFL
EHMT1/GRIN1	MAAADAEAVPARGEPQQDCCVKTELLGEETPMAADEGSAEKQAGEAHMAA...	30	214	244	VYAILVSHPPTPNDHFTPTPVSYTAGFYRI
EHMT1/GRIN1	MAAADAEAVPARGEPQQDCCVKTELLGEETPMAADEGSAEKQAGEAHMAA...	30	214	244	VYAILVSHPPTPNDHFTPTPVSYTAGFYRI
EHMT1/GRIN1	MAADEGSAEKQAGEAHMAADGETNGSCENSDASSHANAAKHTQDSARVNP...	30	183	213	VYAILVSHPPTPNDHFTPTPVSYTAGFYRI
EHMT1/GRIN1	MAADEGSAEKQAGEAHMAADGETNGSCENSDASSHANAAKHTQDSARVNP...	30	183	213	VYAILVSHPPTPNDHFTPTPVSYTAGFYRI
EWSR1/NFATC	MASTDYSTYSQAAAQQGYSAYTAQPTQGYAQTTQAYGQQSYGTYGQPTDV...	30	243	273	RGRGRGGFDRGGMSRGGRGGGRGGMGIPVT

Figure 7. This is the database as visualized in phpmyadmin.

4.1.2.7. User Interface:

The scripts to create the user interface were written in Notepad++. An HTML form was created, with Cascading style sheets(CSS) being used to improve the formatting and

aesthetics; Using HTML 5.0 standards. Following this an action page was written using PHP 3.0 in order to dynamically create a HTML response page to user request. The response page is constructed utilizing Structured Query Language(SQL) queries embedded in the PHP 3.0 script(Please see the Results&Discussion section for sample user interface request and response).

Please note the entire user interface is dynamically generated based on user inputs, middle tier programming, and the backend database.

4.1.2.7. Finding Proteasomal Cleavage Sites on the Linker Regions:

First the list of all proteases, clan-wise (serine, aspartate, cysteine, etc.), were downloaded from MEROPS. Following this all the substrate information namely- substrate name, UniProt ID, Residue range, cleavage site, cleavage type, evidence, P4, P3, P2, P1, P1', P2', P3', P4' amino acids were taken. To obtain tripeptides, tetrapeptides, and pentapeptides motifs for each protease of the Serine clan the residues corresponding to (P3,P2,P1), (P4,P3,P2,P1), (P4,P3,P2,P1,P1') cleavage sites respectively were extracted and saved. A script was written to scour each of the linker region sequences for matches to any of the protease sites. If a match was found, the protease, the cleavage site, the start, and the end point of the matched site on the linker region were all stored in a dataframe which was later converted into an excel file.

This file was then cleaned using the same program which was used to remove duplicates in the linker results file.

Using a python script, we then screen scraped the MEROPS' substrate sites for the residues at the P4, P3, P2, P1, P1', P2', P3', P4' positions for each available protease. The results were stored as a data frame, which can then be used to find amino acid positional frequency. A script was written which could calculate the most frequently occurring residue (or the residue with the highest frequency) at all the positions for the proteases. The frequency for the residues at the positions was calculated as such: the number of times the residue occurs at its respective position to the total count of all the

residues at that position for the protease. This could then be used to further understand the actual probability of cleavage at the site found.

This part of our study remains in progress, and we hope to be able to add this data to the database soon.

4.2. Materials:

4.2.1 Source of Data:

MEROPS: This is a database which is a resource for information on peptidases (Rawlings et al. 1). MEROPS uses hierarchical, structure-based schemes for the classification of the peptidases and inhibitors. Each protease is assigned to a Family on the basis of statistically significant similarities in amino acid sequence, and families that are thought to be homologous are grouped together in a Clan. Aspartic, Cysteine, Glutamic, Metallo, Asparagine, Mixed, Serine, Threonine, Unknown or Compound Peptidase are the main Clans in the MEROPS database.

FusionGDB: Seeing that fusion genes are important players in several major cancer types Kim et al. built FusionGDB (Fusion Gene annotation DataBase) available at https://ccsm.uth.edu/FusionGDB. This database attempts to better understand the function of fusion genes in cancer types, and aid in identifying more clinically relevant fusion proteins. It has a collection of 48,117 fusion genes across pan-cancer from three representative fusion gene resources: the improved database of chimeric transcripts and RNA-seq data (ChiTaRS 3.1), an integrative resource for cancer-associated transcript fusions (TumorFusions), and The Cancer Genome Atlas (TCGA) fusions by Gao et al. (Kim et al. 1). Functional annotations, gene assessment, retention search, of protein features, and ORF assignment was done for all of these proteins in the database. They also provide both the amino acid sequences, and the nucleotide sequences for the fusion proteins according to multiple breakpoints and transcript isoforms. The database provides six categories of annotations: FusionGeneSummary, FusionProtFeature, FusionGeneSequence, FusionGenePPI, RelatedDrug, and RelatedDisease.

Non-redundant protein database and BLAST: When using BLAST the user can align the query sequence against a database of sequences, and place parameters, or constraints in place such as filtering the database for only human sequences. In this study BLAST was used with the non redundant proteins database.The non-redundant database or nr database is one of the database options available, alongside others such as the Swissprot, to use with the BLAST tool. This database is non-redundant, meaning that redundant entries such as multiple entries of a certain viral sequence's strains at the time of a pandemic, where most of them are in fact the same sequence, would be merged into one entry in these databases. To be merged the two sequences must have the same length when compared, and each residue at each position must be identical. The FASTA deflines are not included in this comparison. This database includes all non-redundant GenBank CDS translations +PDB +SwissProt +PIR + PRF excluding environmental samples from WGS projects, it has 294758600 protein sequences at the time of writing(Altschul et al. 403-410).

4.2.2. Programming Tools and Packages:

The OS utilized for the duration of this project was Uniplexed Information and Computing Service(UNIX), Windows 10 only being used when the data was put into MariaDB on XAMPP. The reason behind this was ease of use when programming due to the UNIX terminal, as well as our initial attempt to use Stand alone BLAST being easier to use with a UNIX operating system.

The language used to write the programs for the majority of this project was python, due to the large amount of libraries available to use, as well as my familiarity with the language. I chose python because it was cross-platform as well, and could be ported to UNIX or Windows 10 as required. Pandas and Numpy were notable packages used in this project; when working with large amounts of data, lists and other such data structures can't be used or aren't the most efficient. Therefore we used the Pandas dataframe in conjugation with Numpy arrays to store and work with the data. The data frame was also very useful when working with excel documents since we could easily convert between

the excel file and dataframes. Dataframes allow easy interoperability with Microsoft Excel/Office. Pandas allowed us to append dataframes to one another and put them in excel files, or convert them to csv files as needed, or to edit the large amount of data very easily.

A few other notable mentions are xlrd and xlswriter, which were used to read from and write to excel files, re was used in places where regex was required for pattern matching a substring etc., and lastly itertools was used in order to utilize functions for more efficient looping.

4.2.3 Database Design:

When we decided to create a database before exploring our options we designed important questions which we felt were very pertinent when looking to create a database. We wanted to look at possible users and the data they may want, since that is the most important baseline when designing a data model.

A) Functional Requirements :

- Who will use the data? For now only the members of the lab under which this project was conducted, this means the users will mostly be researchers who might be looking to run simulations, or analyze the linker regions' traits.
- For what?-Diagnosis?, Research?, Cure? This data still needs to be further refined, and will probably only be used for further research for some time. But the end goal is for the data to be used in improving diagnostics and designing better cures.
- What will be the frequency of updates? Pretty frequent, since we're still working on the database.
- Will there be applications (programs) written to access the data? Roughly what will the programs do? This was factored into our choice since we expect this database to be modified, updated, analyzed, and refined further.

B) Non-functional :

- Will there be a lot of users for this data? (scalability) No, this is not something which we need to worry about for the near future.

- Will they need a very high speed of access? Yes.
- Do we need to maintain high security of the data? Not really necessary at this stage.
- Will there be a need to transfer some more data to our db in the future? Or transfer our DB to some other machines? Yes.

After taking these factors into account we narrowed our choices down to two options. MariaDB and MongoDB are both highly popular databases.

MongoDB is a document database, mainly used by people building internet and business applications who need to evolve quickly and scale elegantly. It's available for businesses, but it is also available for free through the open source community edition. This makes it good for personal or small scale use as well.

The main idea behind this type of database is that all data for a single entity needs to be stored as a document and all documents can be stored together within a collection. This document can store the sub-document data, which in RDBMS are typically stored as an encoded string or within a separate table. Each and every document can be accessed by a unique key. Instead of storing data in tables of rows or columns like SQL databases, each row in a MongoDB database is a document described in JSON, a formatting language.

Document databases are extremely flexible, allowing variations in the structure of documents and allowing storage of documents that are partially complete. Fields in a document play the role of columns in a SQL database, and like columns, they can be indexed to increase search performance. Best of all for many developers, the programmer can change the structure of the database easily as needs change. Some say this turns data into code(Saha 1).

According to the ("The Most Popular Database for Modern Apps." *MongoDB*, www.mongodb.com/.) MongoDB is a great choice if you need to:

- Represent data with natural clusters and variability over time or in its structure
- Support rapid iterative development.
- Enable collaboration of a large number of teams
- Scale to high levels of read and write traffic.
- Scale your data repository to a massive size.
- Evolve the type of deployment as the business changes.
- Store, manage, and search data with text, geospatial, or time series dimensions.

MariaDB was created by the original developers of MySQL, who created a commercially supported fork of the MySQL relational database management system due to their concerns when MySQL was being acquired by Oracle. MariaDB is intended to remain a free and open-source software under the GNU General Public License. One of the main pros for MariaDB is that it can be downloaded as a package with XAMPP.

There are two server environments to choose from: local and remote. A local server is, as you might have guessed, hosted locally on your own computer while a remote server is hosted elsewhere. It might be a paid hosting plan, another computer on a local area network, or even a free hosting plan; regardless, a remote server is a server that is not on your computer. As You know that PHP & MYSQL requires Application and Database server to run which is only possible if you have a server machine .

Since we had also decided on creating a web interface for ease of use this was very helpful, since I could create the database in MariaDB and then write PHP,HTML, and CSS to create a website, and test it locally in XAMPP. XAMPP is an application which creates a small server on your localhost computer so that whenever you edit and manipulate your web applications code (Mainly PHP) it helps to show you what your web application will look like .

Chapter 5: Results & Discussion

5.1. Results of initial pilot study:

To begin our study we took 20 fusion proteins as case studies, and tried to isolate their linker regions. During this attempt we discovered that either the fusion proteins could be perfectly fused with no unmatched region at the fusion junction, or there could be a linker region present. This was unexpected since we had hypothesized that an extra region would need to be added to the amino acid code for the protein to be stable and function effectively. Our expectation was that any protein which didn't have such a region would have a truncated domain or some other structural defect and not be able to fuse effectively.

However we discovered later in our research that if the break happened in regions of disorder for both previously ordered proteins, then the fusion could be successful, and the protein could function effectively. We decided to disregard proteins which were fused and had no linkers for our current research, since out of 20 cases we only found 5 such proteins, and since the scope of the current research was limited to investigating only fusion proteins which have linkers associated. We proceeded to design a study to extract data of our region of interest, the added linker region. Please note that the impact or effect of such proteins was not a concern when making this decision.

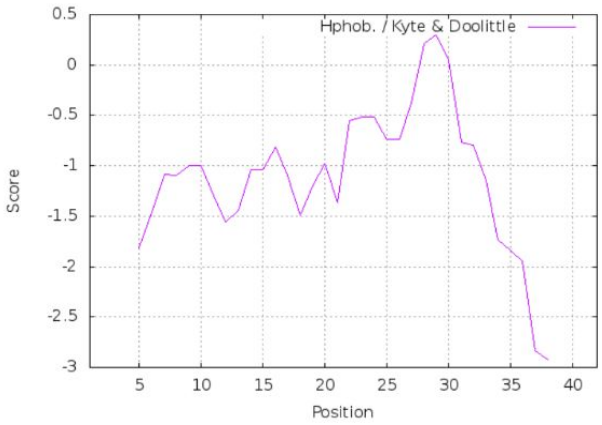
In a few pilot studies we observed more properties of the linker regions found, and tried to get a feel for their traits.

The example in Figure 8 sets the standard for the linkers we found in the 20 cases we took. The linkers we detected from our pilot study were all in highly disordered mobile regions or coil regions according to the disordered region prediction server Disembl. Then we checked their amino acid properties, and how they matched up to definitions of traditional linkers in normal non-oncogenic proteins. This comparison was important to our study since we wanted to better understand if the oncoprotein linkers had any

additional or unusual traits/properties which distinguished them or gave them a functional importance in the protein structure. Normal linker sequences have been observed to avoid large hydrophobic residues to maintain good solubility in aqueous solutions(Xiaoying et al. 1357-1369); This proved true for the fusion protein linkers as well.

>tr|B1PRL2|B1PRL2_HUMAN EWS/FLI fusion protein OS=Homo sapiens OX=9606 PE=2 SV=1
MASTDYSTYSQAAAQQGYSAYTAQPTQGYAQTTQAYGQQSYGTYGQPTDVSYTQAQTTAT
YGQTAYATSYGQPPTGYTTPTAPQAYSQPVQGYGTGAYDTTTATVTTTQASYAAQSAYGT
QPAYPAYGQQPAATAPTRPQDGNKPTETSQPQSSTGGYNQPSLGYGQSNYSYPQVPGSYP
MQPVTAPPSYPPTSYSSTQPTSYDQSSYSQQNTYGQPSSYGQQSSYGQQSSYGQQPPTSY
PPQTGSYSQAPSQYSQQSSSYGQQNPSYDSVRRGAWGNNMNSGLNKSPPLGGAQTISKNT
EQRPQPDPYQILGPTSSRLANPGSGQIQLWQFLLELLSDSANASCITWEGTNGEFKMTDP
DEVARRWGERKSKPNMNYDKLSRALRYYYDKNIMTKVHGKRYAYKFDFHGIAQALQPHPT
ESSMYKYPSDISYMPSYHAHQQKVNFVPPHPSSMPVTSSSFFGAASQYWTSPTGGIYPNP
NVPRHPNTHVPSHLGSYY

ProtScale for linker hydrophobicity(Kyte J., Doolittle R.F.)



Disembl for disorder(coils)

Disordered by Loops/coils definition

>none_LOOPS 18-102, 113-325, 340-379, 389-406, 412-498
mastdystys qaaaqqgYSA YTAQPTQGYA QTTQAYGQQS YGTYGQPTDV SYTQAQTTAT YGQTAYATSY GQPPTGYTTP TAPQAYSQPV QGYGTGAYDT TTatvtttqa syAAQSAYGT QPAYPAYGQQ PAATAPTRPQ
DGNKPTETSQ PQSSTGGYNQ PSLGYGQSNY SYPQVPGSYP MQPVTAPPSY PPTSYSSTQP TSYDQSSYSQ QNTYGQPSSY GQQSSYGQQS SYGQQPPTSY PPQTGSYSQA PSQYSQQSSS YGQQNPSYDS VRRGAWGNNM
NSGLNKSPPL GGAQTISKNT EQRPQPDPYQ ILGPTSSRLA NPGSGqiqlw qfllellsdS ANASCITWEG TNGEFKMTDP DEVARRWGER KSKPNMNYDk lsralryyYD KNIMTKVHGK RYAYKFdfhg iAQALQPHPT
ESSMYKYPSD ISYMPSYHAH QQKVNFVPPH PSSMPVTSSS FFGAASQYWT SPTGGIYPNP NVPRHPNTHV PSHLGSYY

Figure 8. The linker properties of the linker found for the EWS/FLI1 fusion protein. The ProtScale results support the claim that the linker tends to avoid hydrophobic residues, and the Disembl results support the claim that the linker is an intrinsically disordered region and highly flexible.

The linkers seemed to be proving to be very similar to normal linker regions. But in order to make any claims about their sequences and properties we would need a larger dataset. Therefore we set out to make a database.

5.2. Database creation:

Please refer back to Figure 5 which shows the process flow for detecting the linker regions, upon extraction of the linker regions from all the sequences in the FusionGDB database our resulting files had 56.6 MB of data. Upon cleaning and obtaining the final result file, we ended up with 4.62 MB of data. This data was stored in a database as a single table, and a user interface was created to facilitate potential users to access the data. So our final database contained 7,455 fusion protein linkers (Please note this is the largest and only linker database for oncoproteins in the world to the best of our knowledge).

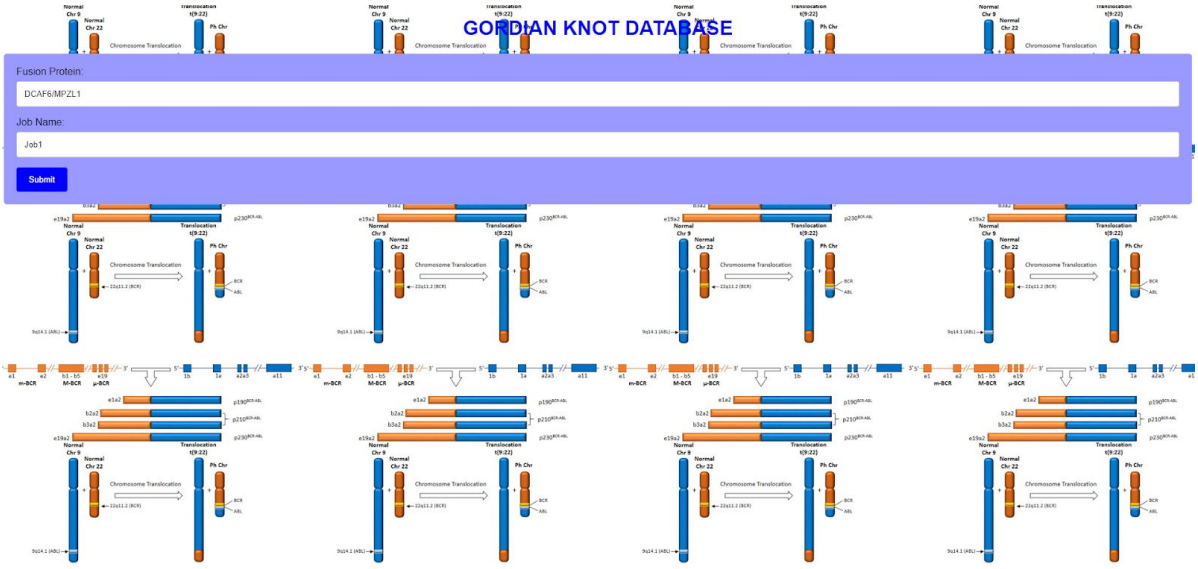
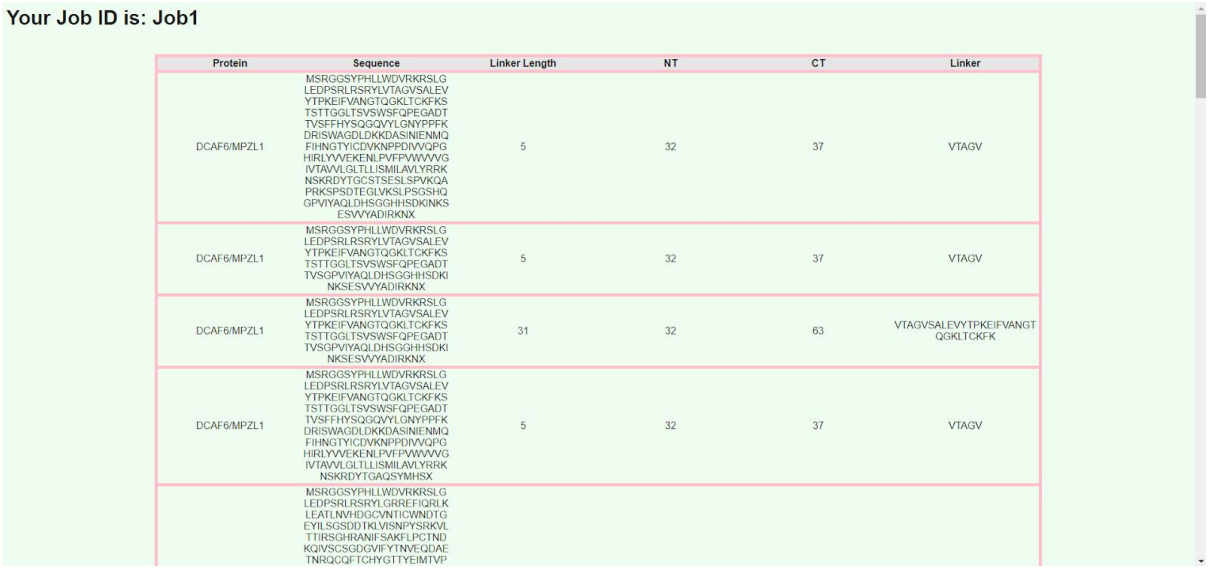


Figure 9. The site's user interface.



Your Job ID is: Job1

Protein	Sequence	Linker Length	NT	CT	Linker
DCAF6/MPZL1	MSRGGSYPHLLWDVRKRSLG LEDPSRLRSRYLVTAGVSALEV YTPKEIFVANGTQGKLTCKFKS TSTTGGLTSVSWSFQPEGADT TVSFFHYSQGQVYLGNYPPFK DRISWAGDLDKKDASINIENMQ FIHNGTYICDVKNPPDIVVQPG HIRLYVVEKENLPVFPVWVVVG IVTAVVLGLTLLISMILAVLYRRK NSKRDYTGCSTSESLSPVKQA PRKSPSDTEGLVKSLPSGSHQ GPVIYAQLDHSGGHHSDKINKS ESVVYADIRKNX	5	32	37	VTAGV
DCAF6/MPZL1	MSRGGSYPHLLWDVRKRSLG LEDPSRLRSRYLVTAGVSALEV YTPKEIFVANGTQGKLTCKFKS TSTTGGLTSVSWSFQPEGADT TVSGPVIYAQLDHSGGHHSDKI NKSESVVYADIRKNX	5	32	37	VTAGV
DCAF6/MPZL1	MSRGGSYPHLLWDVRKRSLG LEDPSRLRSRYLVTAGVSALEV YTPKEIFVANGTQGKLTCKFKS TSTTGGLTSVSWSFQPEGADT TVSGPVIYAQLDHSGGHHSDKI NKSESVVYADIRKNX	31	32	63	VTAGVSALEVYTPKEIFVANGT QGKLTCKFK
DCAF6/MPZL1	MSRGGSYPHLLWDVRKRSLG LEDPSRLRSRYLVTAGVSALEV YTPKEIFVANGTQGKLTCKFKS TSTTGGLTSVSWSFQPEGADT TVSFFHYSQGQVYLGNYPPFK DRISWAGDLDKKDASINIENMQ FIHNGTYICDVKNPPDIVVQPG HIRLYVVEKENLPVFPVWVVVG IVTAVVLGLTLLISMILAVLYRRK NSKRDYTGAQSYMHSX	5	32	37	VTAGV
	MSRGGSYPHLLWDVRKRSLG LEDPSRLRSRYLGRREFIQRLK LEATLNVHDGCVNTICWNDTG EYILSGSDDTKLVISNPYSRKVL TTIRSGHRANIFSAKFLPCTND KQIVSCSGDGVIFYTNVEQDAE TNRQQFTCHYGTTYEIMTVP				

Figure 10. An example output.

5.3. Mean and Standard Deviation of Linker Length:

Similar to our previous case studies, we ventured to find the traits of this larger dataset of linker regions. The average length of a linker can be indicative of its possible functions; Generally modifications in linker lengths has been shown to affect protein stability, folding rates, and since linkers' main function is to connect domains their length has been observed to affect domain–domain orientation(Robinson and Sauer 5929–5934). Thus we endeavored to find the average linker length of our dataset. The linker lengths in our database were summed and divided by the total number of amino acid residues(264,851 amino acids were the total amino acid residues present in our dataset). Upon calculation the average linker length was found to be 35.61(2 d.p.) or 36 residues when rounded off. In an effort to further clarify the average linker length we broke the data into sets of 30.

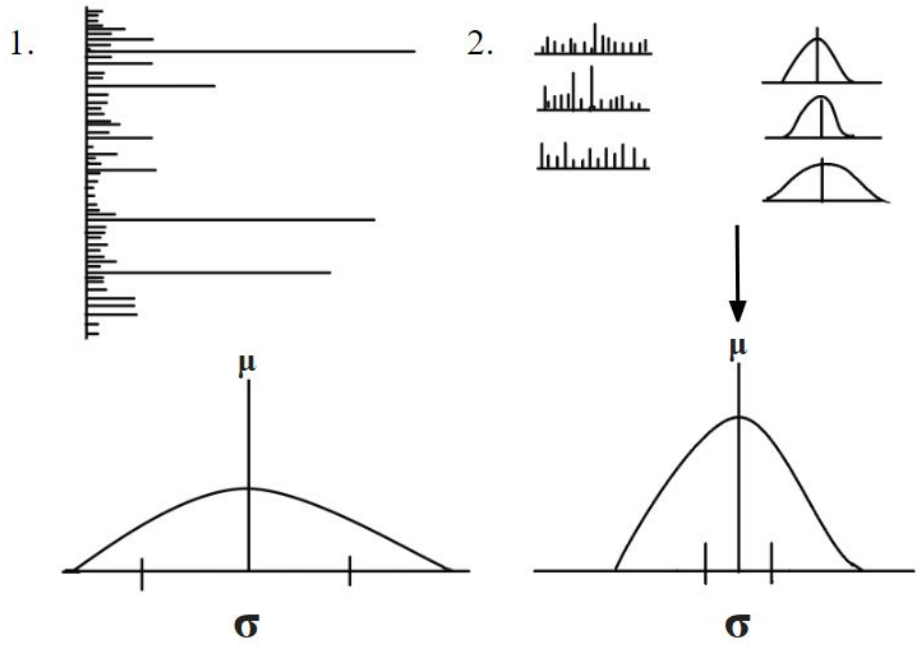


Figure 11. A visual to explain the calculations. Case 1 shows the mean and standard deviation obtained when the sample size is 7455, with the stick graph representing the sample and the graph on the bottom left demonstrating the distribution obtained with this sample. Case 2 represents the distribution we get when we take samples of 30, and how the distribution(seen bottom right) is a lot more normalized when compared to Case 1.

Figure 11 displays a visual representation as to why taking smaller sample sizes may lead to a more accurate representation of the data. Case 1 is taking the entire dataset which has a sample size of 7455 linker lengths, while Case 2 is taking randomly selected sets of sample size 30. We chose 30 because the central limit theorem states that if you have a population with mean μ and standard deviation σ and take sufficiently large random samples from the population with replacement , then the distribution of the sample means will be approximately normally distributed. This will hold true regardless of whether the source population is normal or skewed, provided the sample size is sufficiently large (usually $n \geq 30$).

We used a short script to calculate the average standard deviation of the sample standard deviations and got the value 3.833. The sample means' averages came to 35.630 which was very close to the population average which came to 35.615. Contrarily the population standard deviation was 25.369, this indicates that there is a large deviation in the average linker size in the database. However we feel that this may not be the most accurate representation. Since looking at the data like a population of *all* oncoproteins would not be mathematically correct and not give us accurate results, and since outliers haven't been accounted for; In order to properly visualize the true average and standard deviation it would be better to take smaller sample sizes, and use the sample mean and standard deviations to judge the true spread and average of the data.

Argos calculated the average length of linkers in natural multi-domain proteins to be 6.5 residues; George and Heringa calculated the average length of linkers in natural multi-domain proteins to be 10.0 ± 5.8 residues. George and Heringa further grouped their linker lengths into three groups, small, medium, and large linkers with average length of 4.5 ± 0.7, 9.1 ± 2.4, and 21.0 ± 7.6 residues, respectively. There was a significant difference in the dataset size between Argos' study and George and Heringa's studies. Argos's study only examined 51 linkers while George and Heringa worked with 1280. But regardless of the dataset size we can clearly see that the average linker length for the fusion junction regions is far larger when compared to the average length of linkers in natural multi-domain proteins.

5.4. Linker Length and its Relationship to Protein Structure:

In a study conducted by Ruiz et al. it was examined how the linker region could affect the structure and function of domains, by using cellulase 5A from *Bacillus subtilis* (BsCel5A) as a model. When longer linker regions were inserted between the domains, linker of length 56 residues and 104 residues, the protein was observed to preferentially adopt more condensed structures rather than fully unfolded structures. In a study conducted by Brosey et al. an analysis was performed of replication protein A, a modular multi-domain protein, which has five structural modules connected by long flexible

linkers. Interestingly, Guinier analysis yields an R_g value of 38.8 Å, substantially smaller than that back-calculated from models in which the linkers between domains are fully extended (~53 Å). The study states that this analysis suggests that the longer linker regions lead to the RPA, which can adopt several architectures including those which are uncondensed, especially favoring those that are more condensed than fully extended.

George and Heringa found that the longer linkers showed higher solvent accessibility. The study also found that the average hydrophobicity decreased with the increase of linker length. This indicated that the longer linkers were hydrophilic, and thus more accessible to the aqueous solvent in comparison to shorter linkers. These kinds of insights can prove invaluable when trying to understand the linkers' structure and function, such as the linker length being known to have a higher probability to have structure. Thus we believe knowing the average linker length and linker composition may be important when studying linker traits in an effort to learn more about them.

Two other important traits we investigated were amino acid frequency and amino acid propensity. Argos found Thr, Ser, Pro, Asp, Gly, Lys, Gln, Asn and Ala (in order of decreasing preference) to be the preferred residues for linkers in his small dataset of 51 proteins. He concluded that the preferred linker amino acids are mostly hydrophilic, often polar, and usually small. George and Heringa, with their larger dataset of 1280 proteins found the preferred linker amino acids to be Pro, Arg, Phe, Thr, Glu and Gln, in order of decreasing preference.

5.5. Amino Acid Propensity:

$$Pa = \frac{Nr_{i,l} / \Sigma_i Nr_{i,l}}{Nr_{i,t} / \Sigma_i Nr_{i,t}}$$

Credits: (George and Heringa 871-879)

Figure 12. The formula used to calculate Amino acid propensity. *Pa* is the propensity for an amino acid *i*, $Nr_{i,l}$ and $Nr_{i,t}$ are the number of amino acids type *i* in the linker set (*l*) and in the full protein set (*t*), respectively. $\Sigma Nr_{i,l}$ and $\Sigma Nr_{i,t}$ are the total number of amino acids in the linker set and in the full protein set, respectively.

George and Heringa used a $\chi 2$ test to analyse the significance of trends of the amino acid linker composition between the various linker sets; The sets representing linkers by the number required to connect two domains have no significant compositional differences tested at the 0.1% significance level and therefore suggest that there are no additional amino acid requirements that define the number of linkers connecting two domains.The paper finds that amino acids vary quite a bit in long linkers when compared with short and medium linkers. George and Heringa conclude that the greater the length of a linker is, the more its composition becomes like inter-domain segments. This is supported by our results for amino acid propensity.

Table 1. Amino Acid Propensity Calculation Process.

Three Letter Code	One letter Code	Count(Whole protein)	Mean	Count(Linker)	Mean	Amino acid propensity
CYS	C	121643	0.018	4947	0.019	1.06
ASP	D	333622	0.05	14107	0.053	1.06
SER	S	599765	0.09	24393	0.092	1.02
GLN	Q	337035	0.051	13197	0.05	0.98
LYS	K	406746	0.061	16387	0.062	1.02
ILE	I	291148	0.044	11799	0.045	1.02
PRO	P	434263	0.065	17072	0.064	0.98
THR	T	392488	0.059	15493	0.058	0.98
PHE	F	222089	0.033	9190	0.035	1.06
ASN	N	249495	0.038	10247	0.039	1.03
GLY	G	418170	0.063	16952	0.064	1.02
HIS	H	165653	0.025	6434	0.024	0.96
LEU	L	615174	0.093	23187	0.088	0.95
ARG	R	368087	0.055	14384	0.054	0.98
TRP	W	67406	0.01	2915	0.011	1.1
ALA	A	455131	0.069	17443	0.066	0.96
VAL	V	388829	0.059	15589	0.059	0.98
GLU	E	486742	0.073	19843	0.075	1.03
TYR	Y	137886	0.021	5156	0.019	0.9
MET	M	147551	0.022	6116	0.023	1.05

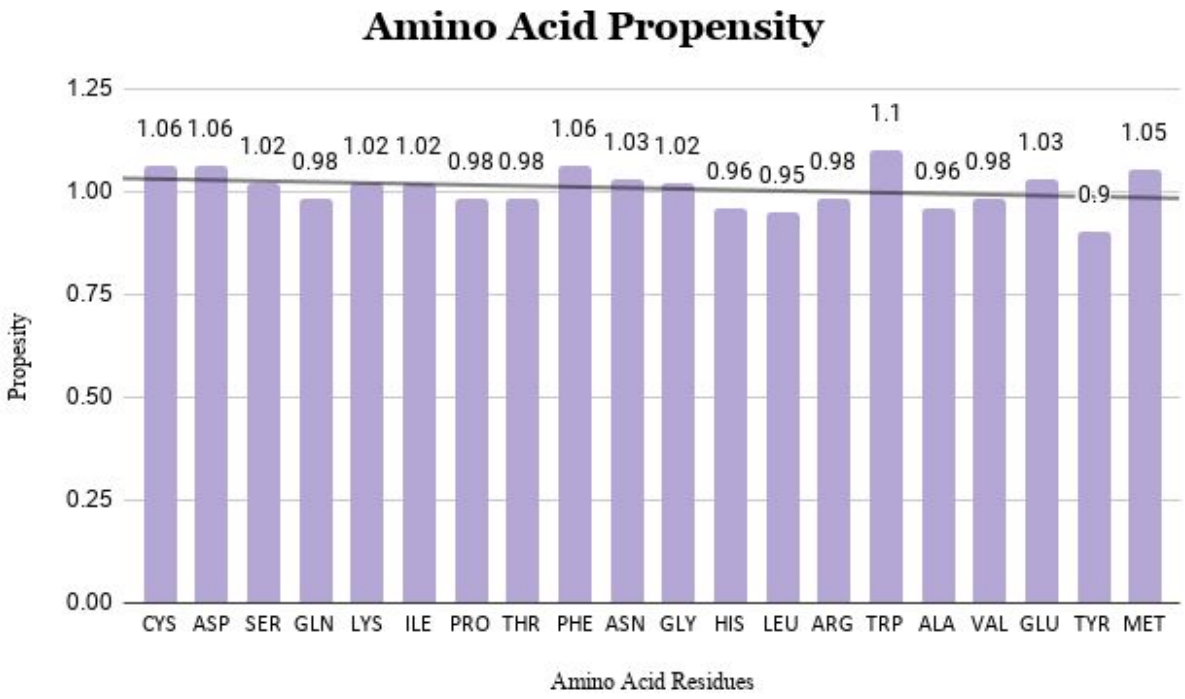


Figure 13. This figure shows a graphical representation of the amino acid propensity values for each amino acid.

As seen in Figure 13 the amino acid propensity values are more or less the same with the trendline showing that each amino acid propensity can be rounded off to 1, the difference in propensities being almost negligible. We believe that due to our larger and more contextual dataset of 7455 linkers, in comparison to the 1280 linkers in George and Heringas' database, as well as the larger average linker length in our dataset, our results were not skewed towards any particular amino acid. In addition to this we believe that the propensity formula's results in George and Heringa's study don't account for the types of proteins in the 1280 proteins in their dataset. The propensity formula is simply comparing the ratios of average occurrence of an amino acid i in the total protein sequences to the occurrence of i in all the linker protein sequences. Due to our proteins being fusion proteins, and not only fusion proteins but also *oncoproteins* we believe that the discrepancy in our results is very significant.

A speculation that may explain these results is that the overall fusion protein is also highly disordered. As stated in the earlier chapter chimera proteins are highly disordered, thus the higher preference for disordered amino acids in a linker region would not be observed in these proteins like it would in normal ordered proteins vs their linker regions. This led us to doing secondary calculations where we found the percent frequency of each residue against only the linker dataset, without looking at the overall proteins sequence. We believe that this could lead to a more accurate visual to the true preference of linkers for certain amino acid residues.

5.6. Percent Frequency of Amino Acids:

Table 2. Calculation Process for the Percent Frequency of Amino Acids.

Amino Acid (One letter code)	Amino Acid (Three letter code)	Count	Percent Frequency
C	CYS	4947	1.87
D	ASP	14107	5.33
S	SER	24393	9.21
Q	GLN	13197	4.98
K	LYS	16387	6.19
I	ILE	11799	4.45
P	PRO	17072	6.45
T	THR	15493	5.85
F	PHE	9190	3.47
N	ASN	10247	3.87
G	GLY	16952	6.4
H	HIS	6434	2.43
L	LEU	23187	8.75
R	ARG	14384	5.43
W	TRP	2915	1.1
A	ALA	17443	6.59
V	VAL	15589	5.89
E	GLU	19843	7.49
Y	TYR	5156	1.95
M	MET	6116	2.31

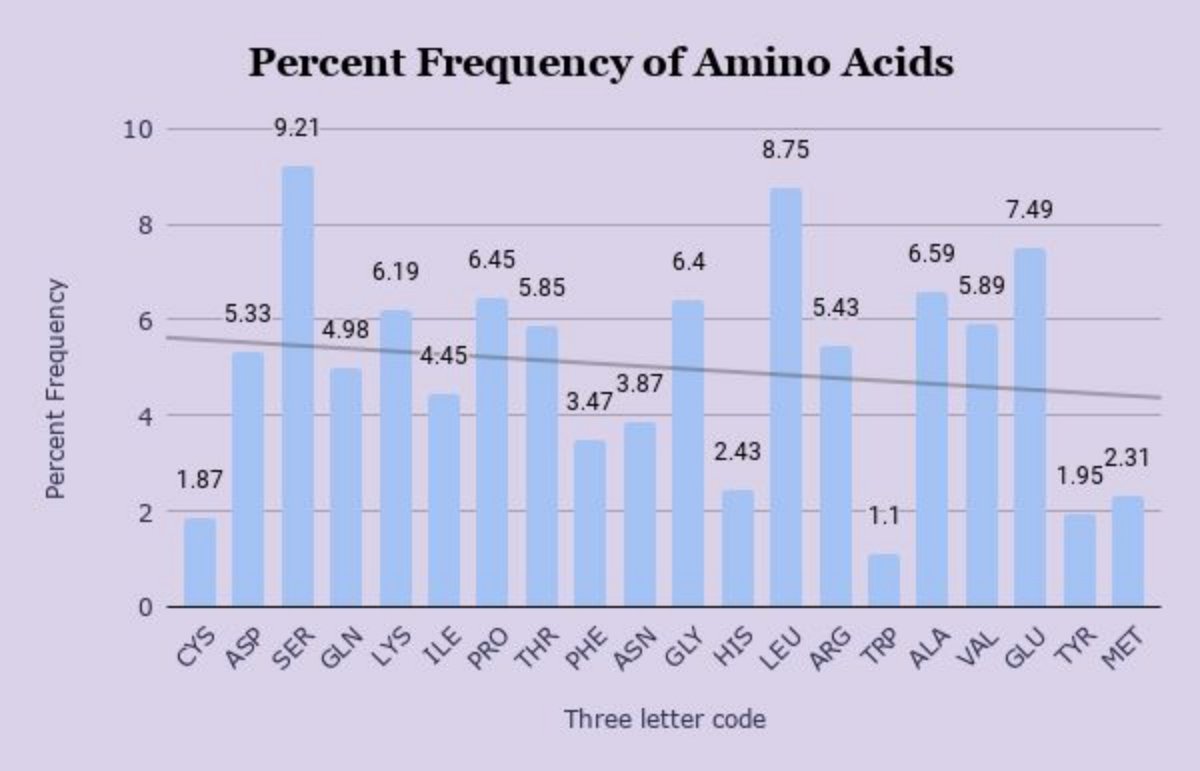


Figure 14. The percent frequency of all the residues in comparison to the total linker residues in the database visualized as a graph.

The percent frequency was calculated by taking the ratio of the total number of occurrences of each amino acid to the total number of linker amino acid residues in the database, and this showed a greater difference in comparison to the amino acid propensity calculations. The trend line still shows that the percentage frequency of the residues doesn't vary all that much, however we can still see some significantly different frequencies.The amino acids highlighted in blue in Table 2 are confirmed to be liked by linker sequences, with the only anomaly Leucine being highlighted in red. Most of the amino acids found agree with Argos' hypothesis that the preferred linker amino acids are hydrophilic, often polar, and usually small.

5.6.1. The High Percent Frequency of Leucine:

We had a few speculations as to why Leucine, which was the amino acid with the second highest frequency, was present in linker sequences. As mentioned earlier in this paper several studies state that linkers avoid hydrophobic residues, and Leucine has not been seen in large numbers in most linker studies conducted such as those of Argos or George and Heringa. We believe that the disordered regions of oncoprotein linkers have a higher percent frequency of Leucine due to their function.

Normally hydrophobic residues induce structure in the protein sequence, since they would logically want to fold and not be exposed to solvent. This led us to hypothesize that the clusters of hydrophobic residues we observed in linker regions may mean that the disordered region folds and becomes ordered, it may mean that there's an ability to bind and have a secondary structure. This could be supported by the fact that most of the oncoprotein linkers are longer than normal protein linkers, meaning there may be a higher probability of structure being present in the sequence.

We saw some proof of this when we aligned the linker sequences Figure 15.

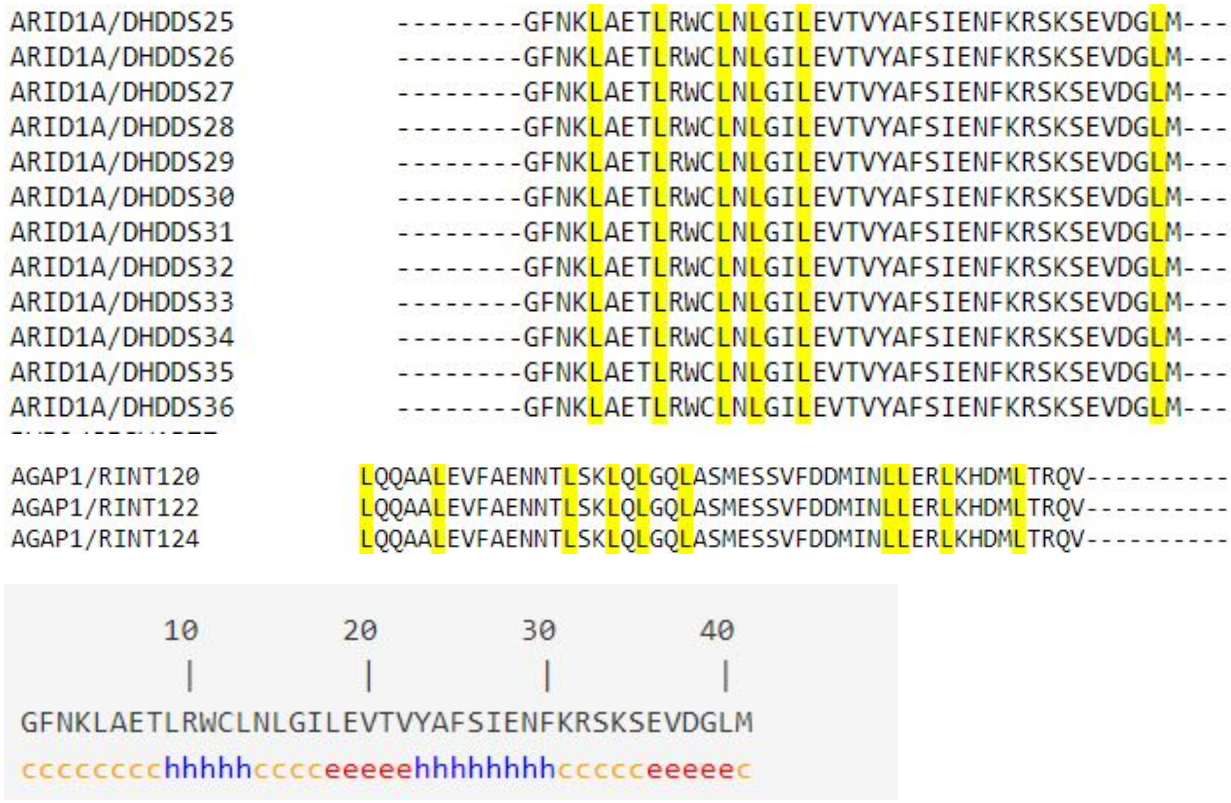


Figure 15. An alignment of linker regions done using ClustalW in an effort to find patterns of any kind in the sequence. Gor prediction for the secondary structure of ARID1A/DHDDS25 was also found and the linker was found to possess helical regions.

These patterns could be indicative of some structure to the sequence, similar patterns are seen in sequences with leucine zippers, and we believe there could be some significance to this, and that further study on the secondary structure of all the linker sequences and amino acid propensities is warranted. These studies may lead to interesting findings and further our understanding of the function of the linker regions.

5.7. Important and Interesting Points to Fuel Further Research:

As discussed in the Literature Review chapter, the following bears repetition: Proteins which are damaged or unneeded are generally degraded by proteases; These are enzymes which break the peptide bonds within the protein, rendering them non-functional. These proteases are responsible for cell cycle control, response to cellular stress, and also play important roles in the immune system.

According to our studies, and the literature we have reviewed we have a few interesting points that warrant further discussion.

1. Linker sequences are unfolded and lack hydrophobic residues, this is important to allow them to function successfully in a protein. This means that such regions being in a protein don't lead to the protein being labelled as misfolded and being degraded. When fusions take place in such a region there is minimal structural disturbance(Uversky 343–384), and it seems likely that since no truncated domains are present the proteins' semblance is very normal, and therefore they are not targeted for degradation.

2. Looking at several cases in literature such as the case by Chris et al., as well as extensive studies on linker properties and functions such as those by George and Heringa, as well as the ones by Argos we believe that the fusion protein linkers play vital roles in chimera proteins. The linkers may act as conduits to maintain domain organization, as well as playing an important role in domain communication so that the protein has the ability to perform a function post fusion. In addition to this if they play the aforementioned roles in a protein then its only logical that they also have an important part to play in the chimera protein's interactions with other proteins. Since several fusion proteins play vital roles in cell control, and signalling pathways the linker region seems to be a significant area to study and target when looking for cancer treatments.

3. Thirdly we hypothesize that the linkers have been evolutionarily designed because autonomy of the folding units allows better interaction between domains, better interactions with other proteins, and aids the protein to fold into compact structures.

These points are of particular interest in our study going forward, since the linker regions in our dataset do seem to show several of these traits. We hypothesize that if we identify a proteasomal cleavage site present in the linker regions it will be highly likely to cleave due to the intrinsic disorder in the region. My data could thus expose vulnerabilities that will enable future studies of potential therapies to inhibit oncoprotein function.

5.8. Detection of Proteolytic Sites:

We have conducted a preliminary analysis where we ran a search for 500 linkers in our database against our dataset of Serine proteases extracted from MEROPS. For 500 linkers we got 10,275 hits of proteolytic sites for the linker regions, this is statistically highly significant and has a large impetus when we explore the curative aspects of future linker studies. An example of the dataset obtained can be seen for one protein in Table 3.

Table 3. The tetrapeptide sites, and protease found to match from the Serine protease dataset with the sequence of MAPK14/MICU2's linker region.

MAPK14/MICU2	trypsin 1	VATR
MAPK14/MICU2	cathepsin G	GLAR
MAPK14/MICU2	trypsin 1	GLAR
MAPK14/MICU2	complement component C2a	GLAR
MAPK14/MICU2	complement factor Bb	GLAR
MAPK14/MICU2	mannan-binding lectin-associated serine peptidase 1	GLAR
MAPK14/MICU2	plasmin	GLAR
MAPK14/MICU2	trypsin 1	LTGR

In Table 3 we see several proteases identified, which can cleave the linker sequence of the linker region found in MAPK14/MICU2. A few of these proteases are even excreted in the cytoplasm and seem to have the ability to be cleaved if they are accessible.

We believe that finding these proteolytic sites for all linker regions, even if they are improbable to be cleaved realistically, is a worthwhile endeavor. Finding these sites is like identifying chinks in armor, and we predict that at least a few of these vulnerabilities can be used in order to improve cancer treatment. It's possible that these linkers could be regulated by processing with proteases, if they are found to be susceptible and we can identify sites which are exposed and accessible there is a possibility that we can engineer proteins, or small molecules to selectively inhibit the fusion proteins. This can be done even with suboptimal sites. The huge benefit of such treatments is that the therapy has potential to replace the more toxic treatment alternatives (with their ensuing future side effects) in cancer. In addition to this biomolecules are being more and more preferred in treatment due to their inherent trait of normally not raising antibodies. Such a discovery could have a profound impact on cancer research and treatment.As we endeavor to continue to accrue big data on Fusion proteins and their linkers we hope to continue making significant discoveries as we mine to gain better knowledge of their structures, sequences, function, and behaviors. One can truly say as far as the future of oncogenics is concerned-in big data we trust.

Chapter 6: Future Scope of Study

1. Firstly we hope to classify the functions of each parent protein, and include these classifications within the database. By gaining better understanding of their functions we can deduce the function of the linker region and see the big picture more clearly. In a preliminary study where we used the UniProt ID retrieval tool to retrieve the Gene Ontology results for all genes in our database, we found a very large and diverse range of functions. This leads us to believe that adding this dimension in our database will add a lot more value to our data.
2. We also hope to clean our data and observe the results obtained when we parse the data we receive from BLAST more stringently. If the difference is significant then we will replace the current data with the more stringently filtered data; or offer users the option to only retrieve data from the more stringently filtered set of linkers.
3. In the future we wish to expand our database by collecting more fusion proteins from other databases such as UniProt, COSMIC, and other sources.We hope to grow the linker datasets we have and accumulate large amounts of data on them. Following this a big data warehouse can be created in order to further analyze trends, patterns, as well as use machine learning to be able to find cures, diagnosis, synthetic applications in building linkers in recombinant proteins. As well as discover areas for future research which can be further studied with wet lab work.
4. We hope to further study the secondary structure of linker regions, and include this information in our database. This can aid us in:
   understanding sequence-structure relationships, visualizing the structure, developing proteins to inhibit the function of the linker region, and may have many other applications.
5. Create a comprehensive website to aid the aforementioned work.

Works Cited

1. Mitelman F, Johansson B, Mertens F. The impact of translocations and gene fusions on cancer causation. *Nat Rev Cancer*. 2007;7(4):233-245. doi:10.1038/nrc2091
2. Chung, Man Ki, and Bo Young Kim. "Recent Updates in Cancer Immunotherapy." *Korean Journal of Otorhinolaryngology-Head and Neck Surgery*, vol. 58, no. 7, 2015, p. 449., doi:10.3342/kjorl-hns.2015.58.7.449.
3. Starks, Marques. "Molecular Immunology." *Immunology and Animal Biotechnology*, Scientific e-Resources, 2019, pp. 94–95.
4. Mertens, Fredrik, et al. "The Emerging Complexity of Gene Fusions in Cancer." *Nature Reviews Cancer*, vol. 15, no. 6, 2015, pp. 371–381., doi:10.1038/nrc3947.
5. Lodish H, Berk A, Zipursky SL, et al. Molecular Cell Biology. 4th edition. New York: W. H. Freeman; 2000. Section 24.2, Proto-Oncogenes and Tumor-Suppressor
6. Johnson, Kirsten M et al. "Identification of two types of GGAA-microsatellites and their roles in EWS/FLI binding and gene regulation in Ewing sarcoma." *PloS one* vol. 12,11 e0186275. 1 Nov. 2017, doi:10.1371/journal.pone.0186275
7. Boulay, Gaylor et al. "Cancer-Specific Retargeting of BAF Complexes by a Prion-like Domain." *Cell* vol. 171,1 (2017): 163-178.e19. doi:10.1016/j.cell.2017.07.036
8. Riggi, Nicolò et al. "EWS-FLI1 utilizes divergent chromatin remodeling mechanisms to directly activate or repress enhancer elements in Ewing sarcoma." *Cancer cell* vol. 26,5 (2014): 668-681. doi:10.1016/j.ccell.2014.10.004
9. Latysheva, Natasha S., and M. Madan Babu. "Discovering and Understanding Oncogenic Gene Fusions through Data Intensive Computational Approaches." *Nucleic Acids Research*, vol. 44, no. 10, 2016, pp. 4487–4503., doi:10.1093/nar/gkw282.

10. Joseph, Agnel Praveen et al. “Cis-trans peptide variations in structurally similar proteins.” *Amino acids* vol. 43,3 (2012): 1369-81. doi:10.1007/s00726-011-1211-9
11. Babu, M. Madan. “The Contribution of Intrinsically Disordered Regions to Protein Function, Cellular Complexity, and Human Disease.” *Biochemical Society Transactions*, vol. 44, no. 5, 2016, pp. 1185–1200., doi:10.1042/bst20160172.
12. Uversky, Vladimir N. “Wrecked Regulation of Intrinsically Disordered Proteins in Diseases: Pathogenicity of Deregulated Regulators.” *Frontiers in Molecular Biosciences*, vol. 1, 2014, doi:10.3389/fmolb.2014.00006.
13. Van Der Lee, Robin, et al. “Intrinsically Disordered Segments Affect Protein Half-Life in the Cell and during Evolution.” *Cell Reports*, vol. 8, no. 6, 2014, pp. 1832–1844., doi:10.1016/j.celrep.2014.07.055.
14. Amet, Nurmamet, et al. “Insertion of the Designed Helical Linker Led to Increased Expression of Tf-Based Fusion Proteins.” *Pharmaceutical Research*, vol. 26, no. 3, 2008, pp. 523–528., doi:10.1007/s11095-008-9767-0.
15. Pierotti MA, Sozzi G, Croce CM. Mechanisms of oncogene activation. In: Kufe DW, Pollock RE, Weichselbaum RR, et al., editors. Holland-Frei Cancer Medicine. 6th edition. Hamilton (ON): BC Decker; 2003.
16. Cooper GM. The Cell: A Molecular Approach. 2nd edition. Sunderland (MA): Sinauer Associates; 2000. Oncogenes.
17. Hait, William N, and Trevor W Hambley. “Targeted cancer therapeutics.” Cancer research vol. 69,4 (2009): 1263-7; discussion 1267. doi:10.1158/0008-5472.CAN-08-3836
18. Nishida, Naoyo et al. “Angiogenesis in cancer.” Vascular health and risk management vol. 2,3 (2006): 213-9. doi:10.2147/vhrm.2006.2.3.213
19. Loeb, Lawrence A et al. “Multiple mutations and cancer.” Proceedings of the National Academy of Sciences of the United States of America vol. 100,3 (2003): 776-81. doi:10.1073/pnas.0334858100

20. Jones, Russell G, and Craig B Thompson. “Tumor suppressors and cell metabolism: a recipe for cancer growth.” Genes & development vol. 23,5 (2009): 537-48. doi:10.1101/gad.1756509
21. Boroughs, Lindsey K., and Ralph J. Deberardinis. “Metabolic Pathways Promoting Cancer Cell Survival and Growth.” *Nature Cell Biology*, vol. 17, no. 4, 2015, pp. 351–359., doi:10.1038/ncb3124.
22. Weinstein, I. B., et al. “Oncogene Addiction.” *Cancer Research*, vol. 68, no. 9, 2008, pp. 3077–3080., doi:10.1158/0008-5472.can-07-3293.
23. Gutierrez, Carolina, and Rachel Schiff. “HER2: biology, detection, and clinical implications.” Archives of pathology & laboratory medicine vol. 135,1 (2011): 55-62. doi:10.1043/2010-0454-RAR.1
24. Baselga, J. “Herceptin alone or in combination with chemotherapy in the treatment of HER2-positive metastatic breast cancer: pivotal trials.” *Oncology* vol. 61 Suppl 2 (2001): 14-21. doi:10.1159/000055397
25. Gsponer, J., Futschik, M. E., Teichmann, S. A., and Babu, M. M. (2008). Tight regulation of unstructured proteins: from transcript synthesis to protein degradation. *Science* 322, 1365–1368. doi: 10.1126/science.1163581
26. Uversky, V. N. (2013). Intrinsic disorder-based protein interactions and their modulators. *Curr. Pharm. Des*. 19, 4191–4213. doi: 10.2174/1381612811319230005
27. Hegyi, Hedi, et al. “Intrinsic Structural Disorder Confers Cellular Viability on Oncogenic Fusion Proteins.” *PLoS Computational Biology*, vol. 5, no. 10, 2009, doi:10.1371/journal.pcbi.1000552.
28. Uversky VN, Oldfield CJ, Dunker AK (2005) Showing your ID: intrinsic disorder as an ID for recognition, regulation and cell signaling. J Mol Recognit 18(5): 343–84.
29. Dyson, H., Wright, P. Intrinsically unstructured proteins and their functions. *Nat Rev Mol Cell Biol* 6, 197–208 (2005). https://doi.org/10.1038/nrm1589

30. H. Mano, "The *EML4-ALK* oncogene: targeting an essential growth driver in human cancer," *Proceedings of the Japan Academy, Series B, Physical and Biological Sciences*, vol. 91, no. 5, pp. 193–201, 2015.
31. Santofimia-Castaño, P., Rizzuti, B., Xia, Y. *et al.* Targeting intrinsically disordered proteins involved in cancer. *Cell. Mol. Life Sci.* 77, 1695–1707 (2020). https://doi.org/10.1007/s00018-019-03347-3
32. Cheng Y, LeGall T, Oldfield CJ, Mueller JP, Van YY, Romero P, Cortese MS, Uversky VN, Dunker AK (2006) Rational drug design via intrinsically disordered protein. Trends Biotechnol 24:435–442
33. Suskiewicz, Marcin J et al. "Context-dependent resistance to proteolysis of intrinsically disordered proteins." *Protein science : a publication of the Protein Society* vol. 20,8 (2011): 1285-97. doi:10.1002/pro.657
34. Andreasson, C. "Regulation of Transcription Factor Latency by Receptor-Activated Proteolysis." *Genes & Development*, vol. 20, no. 12, 2006, pp. 1563–1568., doi:10.1101/gad.374206.
35. Hubbard, S. J., et al. "Assessment of Conformational Parameters as Predictors of Limited Proteolytic Sites in Native Protein Structures." *Protein Engineering Design and Selection*, vol. 11, no. 5, 1998, pp. 349–359., doi:10.1093/protein/11.5.349.
36. Gasteiger E., Hoogland C., Gattiker A., Duvaud S., Wilkins M.R., Appel R.D., Bairoch A.;*Protein Identification and Analysis Tools on the ExPASy Server;*(In) John M. Walker (ed): The Proteomics Protocols Handbook, Humana Press (2005).pp. 571-607
37. Chen, Xiaoying et al. "Fusion protein linkers: property, design and functionality." Advanced drug delivery reviews vol. 65,10 (2013): 1357-69. doi:10.1016/j.addr.2012.09.039
38. [Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. J Mol Biol, 215(3):403–410.

39. [Pearson, 1991] Pearson, W. (1991). Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics, 11(3):635–50.
40. [Pearson, 1995] Pearson, W. (1995). Comparison of methods for searching protein sequence databases. Protein Science, 4(6):1145.
41. [Shpaer et al., 1996] Shpaer, E. G., Robinson, M., Yee, D., Candlin, J. D., Mines, R., and Hunkapiller, T. (1996). Sensitivity and selectivity in protein similarity searches: a comparison of smith-waterman in hardware to blast and fasta. Genomics, 38(2):179–191.
42. Rawlings, Neil D et al. "MEROPS: the peptidase database." Nucleic acids research vol. 38,Database issue (2010): D227-33. doi:10.1093/nar/gkp971
43. Kim, Pora, and Xiaobo Zhou. "FusionGDB: fusion gene annotation DataBase." Nucleic acids research vol. 47,D1 (2019): D994-D1004. doi:10.1093/nar/gky1067
44. "The Most Popular Database for Modern Apps." *MongoDB*, www.mongodb.com/.
45. "Apache Friends - XAMPP 7.4.7 Download". Apache Friends. 2020-06-17
46. R. Linding, L.J. Jensen, F. Diella, P. Bork, T.J. Gibson and R.B. Russell; Protein disorder prediction: implications for structural proteomics; Structure Vol 11, Issue 11, 4 November 2003
47. Robinson,C.R. and Sauer,R.T. (1998) *Proc. Natl Acad. Sci. USA.*, 95, 5929–5934.
48. Argos, P. "An investigation of oligopeptides linking domains in protein tertiary structures and possible candidates for general gene fusion." *Journal of molecular biology* vol. 211,4 (1990): 943-58. doi:10.1016/0022-2836(90)90085-Z
49. George, Richard A, and Jaap Heringa. "An analysis of protein domain linkers: their classification and role in protein folding." *Protein engineering* vol. 15,11 (2002): 871-9. doi:10.1093/protein/15.11.871
50. Ruiz, D., Turowski, V. & Murakami, M. Effects of the linker region on the structure and function of modular GH5 cellulases. *Sci Rep* 6, 28504 (2016). https://doi.org/10.1038/srep28504

51. Brosey, Chris A et al. “A new structural framework for integrating replication protein A into DNA processing machinery.” *Nucleic acids research* vol. 41,4 (2013): 2313-27. doi:10.1093/nar/gks1332

Appendix V

OXFORD UNIVERSITY PRESS | Bioinformatics

The Polybasic Insert of the Covid Spike Protein and the Feline SARS– Evolved or Yet to Evolve

Journal:	*Bioinformatics*
Manuscript ID	Draft
Category:	Original Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Budhraja, Anshul; School of Biotechnology and Bioinformatics, D.Y.Patil University Pandey, Sakshi; School of Biotechnology and Bioinformatics, D.Y Patil University Raghavan Kannan, Srinivasa; Bioinformatics Institute, Agency for Science, Technology and Research ; Nanyang Technological University, School of Biological Sciences; National University of Singapore, Department of Biological Sciences Verma, Chandra; Bioinformatics Institute, Singapore, Biomolecular Modelling and Design Group; Nanyang Technological University, School of Biological Sciences; National University of Singapore, Department of Biological Sciences, Venkatraman, Prasanna; Advanced Centre for Treatment Research and Education in Cancer, Protein Interactome Lab for Structural and Functional Biology; Homi Bhabha National Institute
Keywords:	Bioinformatics, Multiple sequence alignment, Data mining, Drug design, Evolution, Motif finding

The Polybasic Insert of the Covid Spike Protein and the Feline SARS– Evolved or Yet to Evolve

Anshul Budhraja[1,2,3]*, Sakshi Pandey[1,2,3]*, Srinivasaraghavan Kannan[a], Chandra Verma[a,b,c] and Prasanna Venkatraman[2,3]#

[1] School of Biotechnology and Bioinformatics, D.Y. Patil University, Sector 15, Plot No 50, CBD Belapur, Navi Mumbai, Maharashtra 400614

[2] Protein Interactome Lab for Structural and Functional Biology, Advanced Centre for Treatment, Research and Education in Cancer, Sector 22, Kharghar, Navi Mumbai, Maharashtra, India 410210

[3]Homi Bhabha National Institute, 2nd floor, BARC Training School Complex, Anushaktinagar, Mumbai, Maharashtra, India 400094

[a]Bioinformatics Institute, Agency for Science, Technology and Research (A*STAR), 30 Biopolis Street, #07-01 Matrix, Singapore 138671.

[b]School of Biological Sciences, Nanyang Technological University, 60 Nanyang Drive,Singapore 637551.

[c]Department of Biological Sciences, National University of Singapore, 14 Science Drive 4, Singapore 117543.

*Equal Contribution

Corresponding Author

vprasanna@actrec.gov.in

+91 9820097451

Abstract

Recent research on Covid 19 pandemic has exploded around the Furin cleavable polybasic insert PRRAR↓S, in the spike protein [1, 2]. The insert and the Receptor Binding Domain (RBD) are vital clues in the Sherlock Holm-like investigation into the origin of the virus and in its zoonotic crossover. Based on comparative analysis of the whole genome and the sequence features of the insert and RBD domain, the bat and the pangolin have been proposed as very likely intermediary hosts [1-3]. We follow these leads and provide a different perspective on the polybasic insert, its evolution, and the intermediary host. Our investigations suggest that either the SARS Cov2 is in transit on its evolutionary path to gain a fully optimized Furin site or it has promiscuously adapted to the host proteolytic environment. With the selection residues with lowered binding potential at the P5 and P2 positions, the binding mode seems to have been carefully selected posing a challenge to inhibitor design. The polybasic site and the RBD domain seem to have evolved along the lines of Feline SARS spike protein providing substantial evidence for the domestic cat as a possible intermediary host.

Introduction

Among the infectious viruses, mutations that increase receptor affinity and optimize a proteolytic cleavage site invariably enhance virulence [4-6] Therefore, when the novel SARS 2019 virus (CoV2) was spotted to contain a unique polybasic insert on its spike protein readily cleavable by Furin like enzymes [7], coupled with changes in the receptor-binding domain that enhanced affinity [8, 9], everything seemed to fall in line. Recent experiments in a lung cell line using pseudoviruses with the Covid Spike protein support these predictions; mutations or deletions in this region abrogates proteolytic cleavage and prevents infectivity [10, 11]. Here we ask a rather different but crucial question: how abundant or frequent is the occurrence of this polybasic insert of dubious origin in curated protein databases? How does the unique polybasic insert bind to the furin active site?

Results

The polybasic insert of the Novel SARS spike protein is rare among several hundred proteins with RRARS

Furin like enzymes are extremely important for the processing of regulatory proteins inside cells. Viruses have adapted to the human proteolytic environment and the virulent forms are naturally selected for the presence of the polybasic site at the domain boundaries. From the analysis of the collection of such sequences within viral glycoproteins reported in the published literature [4, 12], the sequence PRRARS is rather unique. To obtain an unbiased account of the abundance of PRRARS, we realized that a different approach was needed. We decided to search -non-redundant databases. To the best of our knowledge, there has been no systematic investigation to identify all polybasic possible inserts from databases. Since MEROPS [13] a depository of proteolytic cleavage sites has no record of the P5 residues (Merops deposits octapeptide information), we initially searched nr (non-redundant) human databases for an exact match to PRRAR↓S. To our surprise this particular pentapeptide motif was present only in two human proteins; one, AAB17869.1 Hermansky-Pudlak syndrome protein (HPS1) and the other AAF79955.1 RhoGEF. There are 17 isoforms of the HPS1. RhoGEF in the curated UNIPROT data does not carry the motif. This effectively narrows down the hunt to a single human protein HPS1! (**Supplemental Table 1**).

To ensure that the sequence coverage is adequate, we changed the query sequence and searched human-only database, viral -non-redundant database and the non-human/non-viral databases for a match to 'RRARS' (**Supplemental Table 1**). We plotted the frequency of residues that would occur in the P5 position (**Figure 1**). The same two human proteins were retrieved (2.5% frequency). In all other non-human databases, Proline occurred at a frequency of 5.74%, within the viral database (taxon ID: 10239) at a frequency of 3.79%. We also surveyed an all viral database ViPR for the presence of RRARS and found only two matches- M protein of NL63-related Bat coronavirus, and an Alphacoronavirus from a feline strain (**Supplemental Table 2**). The number of proteins searched and the hits obtained are compiled in **Supplemental Table 3.**

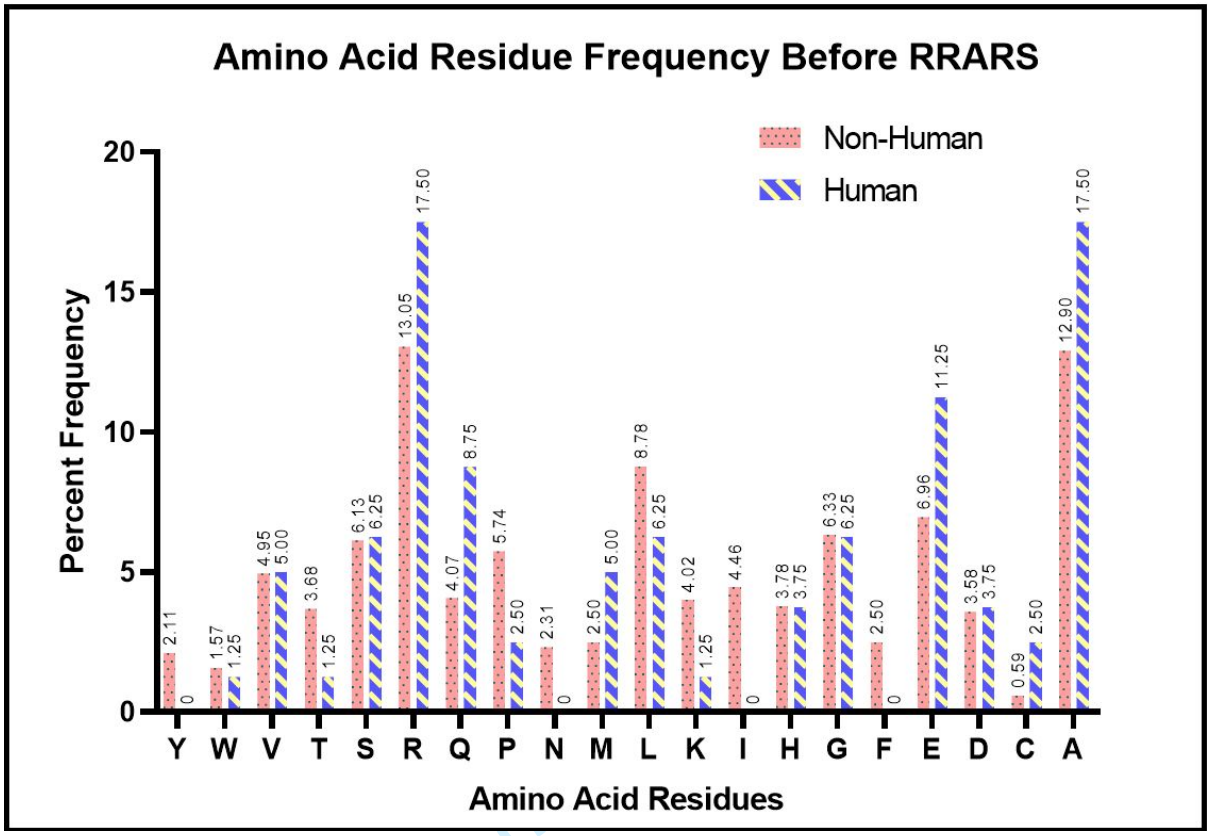


Figure 1 Frequency of distribution of the putative P5 residue in proteins that have the RRAR↓S motif. Proteins carrying the perfect match for this cleavage sequence of Furin and Furin like enzymes RRAR↓S were fetched using the program Blast.

Covid 19 Furin like site shares features of the milder and lethal forms of Feline SARS spike protein

A close variant of the sequence, ARRARS present in a Feline SARS spike protein caught our attention. This is the only coronavirus spike protein with a match for RRARS. Literature studies presented some interesting facts about this Feline spike protein. It was first reported in viral strains that caused mild respiratory and enteric infection in cats (FECV). However, in a small percentage (5%) of the infected cats, a systemic fatal inflammatory disease called the Feline Infectious Peritonitis (FIPV) develops. FECV replicates predominantly in the intestinal epithelium and carries the highly conserved **RRSRR↓S,** one of the best Furin cleavage motifs [14]. In sharp contrast, the FIP spike protein carried mutations at all crucial positions in the polybasic region and many of these mutations adversely affected cleavage by Furin [14]. Apparently, the virus leaves the intestinal epithelium by acquiring cleavage sites for enzymes of the macrophages such as the Cathepsins, MMPs, and PSCK1 accounting for

its expanded tropism. The alignment of FCoV serotype 1 Spike proteins with FIPV serotype (from UniProt P10033) is shown in **Supplemental Figure 1**. The polybasic insert is not seen in the FIPV serotype and one of the FCoV isolate sequences, the insert based on MSA carries P1 Serine, which is unlikely to be cleaved by Furin and Furin-like enzymes.

The presence of a Furin cleavage site ARRAR↓S or RRARR↓S in the Feline spike protein and the presence of PRRAR↓S in the novel SARS prompted us to analyze the sequences more carefully. The all basic RRARR↓S is a bonafide Furin substrate with the sequence satisfying all rules for the position-specific amino acids. However, PRRARS is not. A detailed catalog of amino acids preferred by Furin indicates that Pro is disfavored at the P5 position [15] and yet here we have Pro at P5 of the novel SARS polybasic insert. We already saw that Pro at the putative P5 position is rare among all non-redundant databases! Furthermore, Furin has a stringent requirement for a basic residue (R/K) at P2 position [16] which is an Ala in Cov2. This prompted us to look for the frequency of P2 Ala in cleavage sites of Furin and Furin like enzymes. Ala is one of the low abundant amino acid at this position and is present at 0.03-0.04%

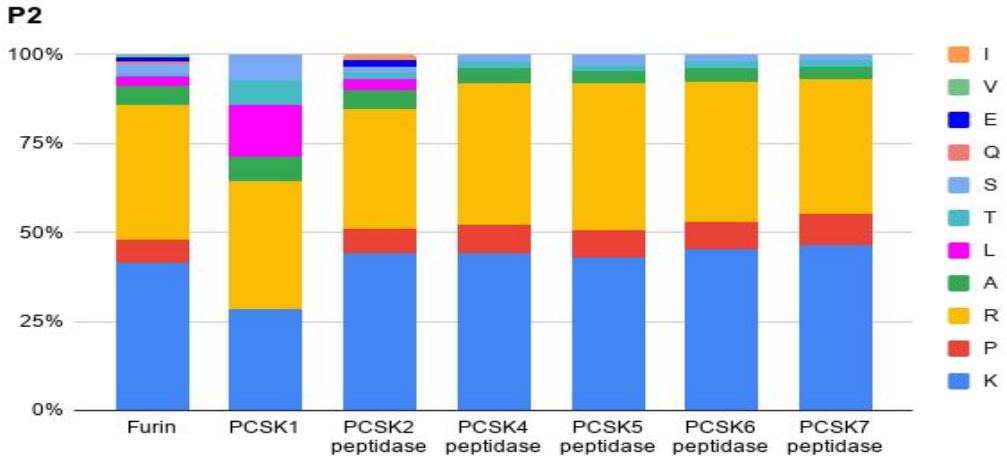


Figure 2: Frequency of amino acid residues found at the P2 position of all the known cleavage sequences of Furin and Furin like enzymes. The cleavage sequences of the form P4P3P2P1P1'P2'P3'P4' archived in the MEROPS database were extracted. Each amino acid present at the P2 position was counted and normalized to the total observed sites. Note that the basic residues R/K contribute to about 80% of the P2 residues.

PRRARS binds less well to Furin as compared to RRKRRS

The possibility that the spike protein could be cleaved by more ubiquitous Furin and Furin-like enzymes as compared to the monobasic arginine at the S1/S2 boundary offered the molecular rationale for the 'severity' of the infection. Recent experiments showed that the polybasic insert in the SARS spike protein can be cleaved both by Furin and a transmembrane protease TMPRS [11, 17]. Respective inhibitors, mutations, or deletions of the sequence deter cleavage and infection. The cleavage was compared with the classical all basic consensus sequence RRKRR↓S by replacing the WT sequence in the Cov2 spike protein was found not to have any exceptional effect. However, antibody-based detection of cleavage products is not quantitative in nature and does not provide information such as the k_{cat} or Km values required for an appropriate comparison. Experiments such as the cleavage of short peptides derived from the Feline spike protein from the different strains by purified Furin as reported in [14] are useful to understand the contribution of position specific amino acids to enzyme efficiency. Alternatively, we can use the power of surrogate methods such as structure-guided modeling and docking to evaluate some of these parameters.

A model of the Furin – RRKRRS an optimized Furin sequence and PRRARS was generated and both the models were subjected to Molecular Dynamics Simulations for further refinement. During MD simulations both the peptides remained stably bound and formed several hydrogen bond interactions with Furin. To further understand the differences between the two sequences that may influence binding affinity, binding energy calculations were carried out. The polybasic sequence (RRKRRS) had a considerably improved binding energy ~ 30 kcal/mol more as compared to the sequence from the SARS Cov 2 protein! All the six residues from the RRKRR↓S complex is involved in hydrogen bond interaction with Furin, whereas in the case of PRRAR↓S sequence, the Proline is not involved in any interactions and alanine interactions are restricted to its backbone (**Figure 3**). While the peptide with residues P5 Pro and P2 Ala in PRRARS remain bound during 100ns simulation, it is the loss of several crucial hydrogen bonds made by P5R and P2K at the binding pocket that costs energy. This huge difference in binding energy is not captured in the experiments such as western blot used to follow the proteolytic cleavage. It is possible that there are other allosteric sites or the secondary exosite recognition by Furin like enzymes may compensate for the rather lower binding energy at the active site. Alternatively, the presence of more protease could compensate for the low turnover anticipated from the poor binding sequence.

Viral or bacterial infections can induce transcription factors such as hypoxia-inducible factor 1 (HIF-1) which enhances the expression of Furin [18]. By virtue of more enzyme, suboptimal basic insert such as the PRRAR↓S eventually could be cleaved.

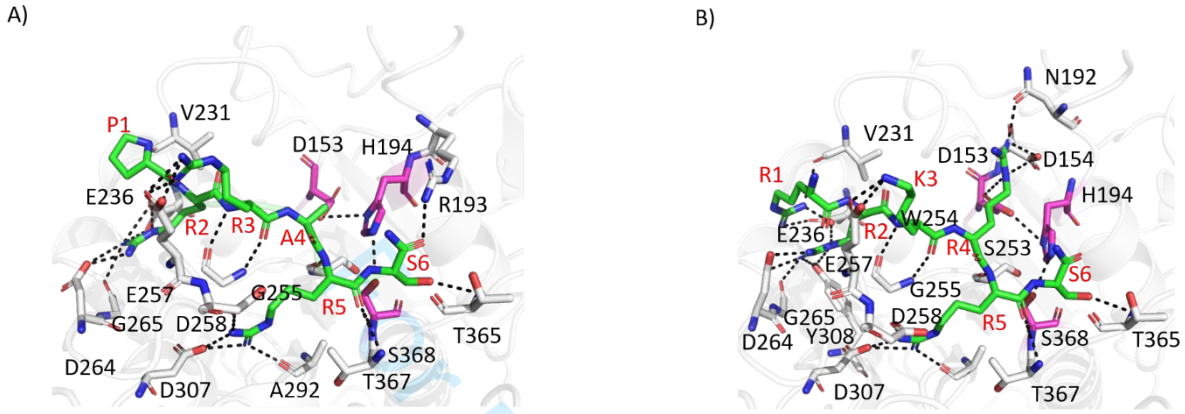


Figure 3: Molecular Dynamics snapshot of (A) Furin-PRRARS (B) Furin–RRKRRS complex. Furin protein is shown as a cartoon (grey) and the bound peptide is shown as a cartoon (green), and peptide-protein-interacting residues and h-bond interactions are highlighted in sticks and dashed lines, respectively. The three catalytic residues (His194, Asn295, and Ser368).

If the sequence is not a very good Furin substrate, then what may be the advantage for the polybasic insert?

As described above the feline SARS with the fully optimized polybasic insert in the spike protein evolved to lose the furin cleavage site. This loss seems to have helped the SARS virus in gaining access to other cell types such as the macrophages in the infected cat. The FPV serotype thus evolved causes fatal systemic infection. Enzymes such as the PCK1 present on the macrophages were expected to cleave the mutant site. This prompted us to ask if the novel polybasic suboptimal Furin site in SARS Cov2 spike protein could be cleaved by other proteases.

We had earlier developed a computational method called the PNSAS, Prediction of Natural Substrates from Artificial Substrates, to predict proteolytic sites within the human proteome [19]. The simple algorithm takes tripeptides of the type P1↓P2P3 or tetrapeptides of the kind

P1'↓P1P2P3 which includes the scissile bond and scans for the presence of the motif within the intended substrate. If a match is found, the algorithm looks for a crystal structure of the protein and if coordinates are found, in the PDB for at least 75% of the protein sequence the surface accessibility is calculated. Subsequently, the accessibility is compared with a completely disordered region and relative accessibility is calculated. Yet another filter is applied for subcellular co-localization upon which the enzyme-substrate relationship is predicted. Prediction from this algorithm was tested on an enzyme and its novel substrate and was authenticated in a breast cancer cell line [20]. The prediction from this algorithm was also validated by two other independent studies [21, 22]. We used the same principles here and integrated different databases for accessibility calculations.

We scanned the dictionary of the cleavage sites of serine proteases for a match with RRARS and RRAR. With the restricted and stringent pentapeptide motif, we precisely identified Furin and the preprotein convertases PCSK 2,4,5,6 and 7 as the only possible enzymes that would cut the polybasic sequence RRAR↓S (data not shown). However, when a tetrapeptide motif was used to increase the possible number of matches besides Furin and Furin like enzymes, human airway proteases matriptase 1 and 2, hepsin and Desc1 aligned with the RRAR (**Figure 4**).

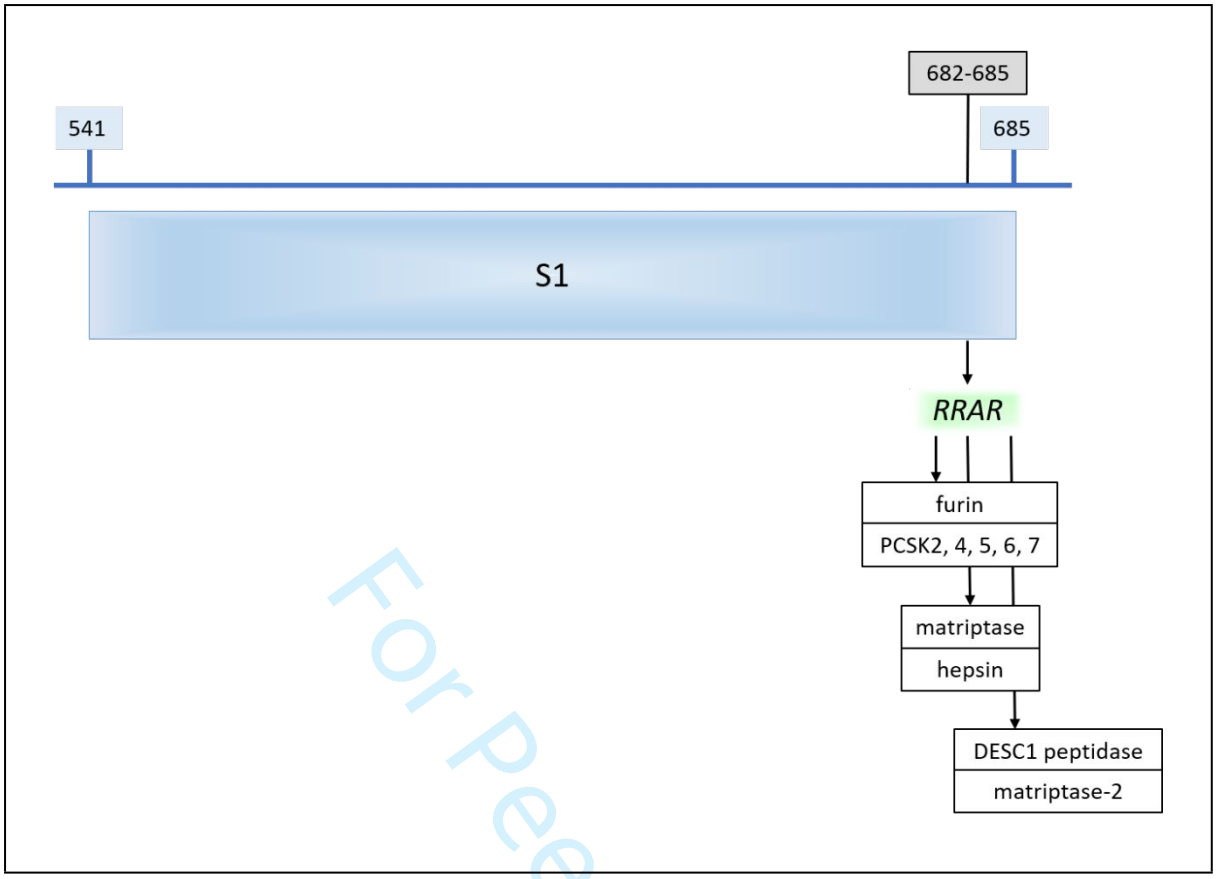


Figure 4. Trypsin like enzymes that can cleave the S1/S2 boundary and activate the virus. A more flexible tetra peptide query set derived from cleavage sites of Serine proteases deposited in MEROPS was used to scan the CoV2 spike protein. The sequence matched sites, accessible on the surface of the S1 domain are indicated.

We calculated the number of times P4R, P3R, P2A, P1R, and P1'S occurred in the cleavage sites of the three Trypsin like enzymes **(Figure 5**). As expected like Furin, these enzymes have a very strong preference for P1R. They have a far better preference for P2 Ala than the Furin like enzymes. They accommodate P1'S and P4 Arg at lower frequency. They have a strong liking for Proline at P4 (disliked by Furin like enzymes) and hydrophobic amino acids such as Ala at the P1'. Assembling these various aspects, we hypothesize that, the Cov2 polybasic insert has retained some of the classic and primary requirements for cleavage by Furin and the Furin-like enzymes i.e., P4 R, P1 R and P1'S residues. This makes Site 1 a better Furin substrate as compared to all other SARS (spike protein) that cause the human infection. However, the P2 Ala and P5 Pro are likely to compromise the efficiency of the Furin and Furin like enzymes.

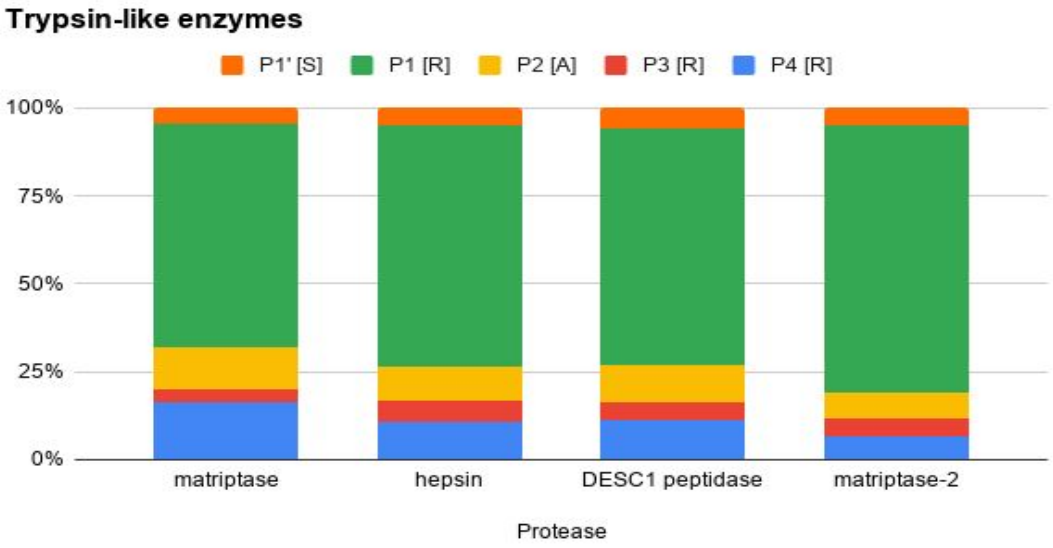


Figure 5. Frequency plot of P4R, P3R, P2A, P1R, and P1'S within the known cleavage sites of the Trypsin like enzymes. The cleavage sequences of the form P4P3P2P1P1' archived in the MEROPS database for the trypsin-like enzymes (X-axis) were extracted. Each amino acid present at these positions were counted. The number of times the amino acids of the polybasic insert RRARS occur were normalized to the total observed sites.

RBD of feline SARS spike protein known to bind to aminopeptidase receptor shares hot spot residues with novel SARS RBD

One of the compelling issues in the current pandemic has been the identity of the intermediary hosts. Depending on whether we are looking at the overall sequence similarity or the sequence conservation at the RBD domain for binding to the human receptor, or the polybasic insert, for example, the possible intermediary host would differ [1-3, 23]. Rampant and large-scale recombination between different lineages of coronaviruses, mutations/insertions/deletions at the S1/S2 site, makes such deductions even more complicated. In addition to the molecular mechanisms required to explain the origin of the virus, there is a practical demand which is, for the originating virus to have acquired both the polybasic insert and the RBD it would have to encounter a host with high population density. Recently the domestic cat is one among the animals that was found to support the replication of novel SARS [24].

The observations from our study clearly indicate that the polybasic insert of Cov2 spike protein has properties close to that of the feline spike protein. The amino acid sequence of Furin from both species is ~92% identical (**Supplemental File 1**). The feline ACE2 receptor is 85.2% identical to the human ACE2 receptor (**Supplemental File 2**). Most of the amino acids of the human ACE2 that are involved in the interaction with CoV2 RBD are conserved in the Feline ACE2 receptor (**Figure 6**). While this has been noted before the RBD of the feline spike protein has not been analyzed for its ability to bind to human ACE2. Moreover, the feline spike protein is only known to bind to aminopeptidase receptors. We compared the two spike protein sequences and find that despite the low sequence homology and distant evolutionary relationship to the novel COVID spike protein (**Figure 7 and Supplemental File 3**), the RBD domain of the feline coronavirus carries at least six residues that are identical to that of the SARC CoV2 seen at the interface of human ACE2-CoV2 RBD protein complex (**Figure 6**). Notable among them are the K417 (not found in SARS CoV) and Tyr 505 found critical for affinity (**Figure 6**). These observations render strong support for the already suspected domestic cat as one of the intermediary hosts that has nurtured the Covid 19 in its zoonotic jump.

Figure 6 Interface of the SARS CoV2 RBD and the ACE2 receptor complex compared with the sequence of the Feline ACE2 and Feline SARS RBD domain. The crystal structure (PDB ID: 6m0j) was used to obtain the interaction map in the 2-dimensional format using PDBsum. Using the pairwise sequence alignment (in Supplementary File 2 – for ACE2 and Supplementary File 3 – for Spike proteins), we juxtaposed the corresponding residues from feline ACE2 and feline SARS RBD domain. Tyr 505, Lys417, Asn501 of the RBD domain

are absolutely conserved. The charged residues Asp30, Lys31, Lys353, Asp355 of ACE2 are absolutely conserved.

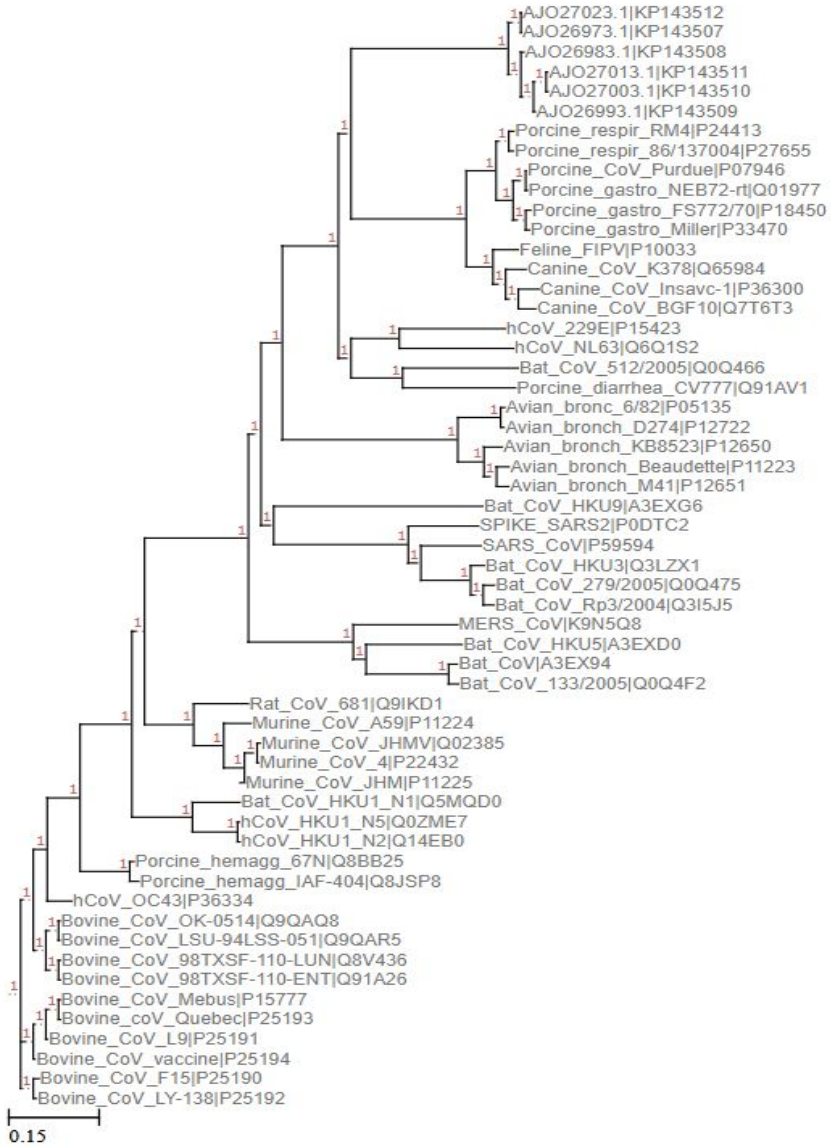


Figure 7 Phylogenetic map of corona viruses based on the sequence homology at the S2 domain of the Spike protein. Curated UniProt sequences of Coronavirus were clubbed with the Feline spike protein sequences reported in [9] and the phylogenetic tree was constructed according to Multiple Sequence Alignment (please refer to method for details).

Discussions

Recently authors of a seminal Letters to Nature put forth several possibilities to explain the origin of the novel SARS 2019 [25]. One among them is the possible identification of a SARS Cov2 like-animal virus with a partial or fully optimized polybasic site. The FECV strain of the Feline virus has a fully optimized Furin cleavage site in the spike protein which is eventually mutated within the infected cat to a lethal serotype called the FPV. Contrary to expectations, this conversion occurs by losing the Furin cleavage capability that seemed to have allowed the virus to cause systemic infection eventually killing the infected cat. This is an important point to ponder about the association of the proteolytic site with the virulence of SARS Cov2 and its evolution. Our analysis show that the polybasic sequence is not one of the optimized sequences for Furin. The P2 Ala and the P5 proline do not engage with the active site suggesting that there is more to the insert than what has been deduced by the nature of the insert so far. Even though Furin and Furin-like enzymes are mandatory for normal functions, they have been proposed as potential targets for inhibitor design. The strategy is to use the inhibitors transiently and in vehicles that can be applied topically. Although the site has not evolved in response to Furin inhibitors, our investigation suggests that it will be very difficult to design inhibitors against the binding pocket to target the spike protein but spare the host enzymes. It is also possible that the novel sars is en route to evolving into an even more virulent strain by acquiring a fully optimized Furin site.

From our analysis, it is also clear that the type II transmembrane human airway trypsin-like epithelial proteases (DESC1, Hepsin, and the Matriptase) can cleave the polybasic site albeit less efficiently. None of these enzymes have been associated with SARS infection so far. However, all these enzymes cleave HA of the influenza virus and this cleavage is associated with infection and virulence [6]. Matriptase, in particular, supports multiple rounds of replication of H9N2 in human airway epithelium [26]. Notably, ubiquitous access of this secreted enzyme to multiple tissues makes it a dangerous ally in breach of organ tropism, virus spread, and pathogenicity of the H9N2. Matriptase bound to the membrane and within the endosomes is known to activate the Influenza A virion (11, 73). Thus, matriptase, by virtue of such subcellular distribution, could potentially be responsible for viral-host membrane fusion during entry and cell-cell fusion during egress. Hepsin and the HAT like

enzymes cleave the cryptic S2' site on the spike protein, however,their role in infection is unclear.

Besides optimal active site geometry and extensive interactions with the substrate, other factors point out to a more complex picture regarding cleavability of the polybasic insert and the S2' site. Three O linked glycosylations are potentially possible in and around this region and these seem to have been' inherited' along with the new insert [25]. The pattern of glycosylation and their branching is dictated by enzymes in the secretory pathway which have very different distribution depending on the cell type. Glycosylation can render the site inaccessible to a protease (or an antibody). Therefore, when and where the cleavage site gets exposed or whether additional deglycosylating enzymes need to be hijacked are some of the relevant questions in the proteolytic activation of the novel SARS.

The S2' site (P)KR↓S is both partially buried in the crystal structures and is flanked by two Asn residues one of which has the strong potential to be glycosylated (Supplemental **Table 4**). Asn at 801 is glycosylated as seen in the EM structures of Cov2 Spike protein (PDB) prepared from HEK 293 cells. The sugar moiety is located 18Å away from the proteolytic site and is unlikely to cause steric hindrance. Although cleavage at this site is detectable only in minor amounts as in the case of SARS Cov [27] or in the presence of excess trypsin, it is adequate enough to cause membrane fusion. The enzymes associated with the S2' cleavage do not prefer Ser at P1'. Therefore, mutations that enhance proteolysis at the S2' site are likely to be far more potent in enhancing virulence.

Glycosylation is a cell type-dependent phenomenon. Whether the glycosylation pattern is the same or different in the relevant lung epithelial cells/airway epithelium and whether it may impact proteolysis remains to be seen. If more branched sugars (in lung cells for example) mask the S2' cleavage site the virus may evolve by mutating the Asn to Asp and this may render the virus more virulent as reported for H9N2 [18] and increase its geographical and ethnic boundaries.

In summary, the spike protein of the novel SARS presents a complex picture both at the level of the polybasic insert and of the Receptor binding domain. It seems plausible that many tricks of the trade could have been selected for by the environmental adaptation in this simple domestic cat. It really seems a smart evolution where the insert is both specific to Furin like

enzymes and mosaic/promiscuous for the Trypsin like enzymes in the airway and a more ubiquitous enzyme such as the matriptase. Many more basic studies that can readily compare the catalytic efficiencies of Furin and Furin like enzymes, information on the exotic , allosteric site, and other conformational changes are needed for a thorough understanding of the potency, precise role and strength of the polybasic insert in the virulence. The worrisome picture is the prospect that mutations and further optimizations of the proteolytic sites at the domain boundaries are possible which may render the virus more virulent and dangerous, thus pre-empting yet another pandemic.

Methods

Blast search to identify proteins carrying the CoV2 polybasic insert A protein blast search for the RRARS motif within the nr database was conducted. This search excluded Homo Sapiens (taxon id: 9606). The maximum allowed number of outputs was 20,000 and we retrieved 15768 hits with 100% identity and 100% coverage for the non-human sequences. From the blast results, file accession numbers of all available sequences were extracted and saved into a text file. Then, using batch Entrez, the corresponding sequences for the accession numbers were fetched and downloaded in the FASTA format. 241 human sequences with 100% identity and 100% coverage were retrieved in an independent blast and the output went through the same pipeline. To increase the coverage and overcome the limitation set by BLAST for searchable sequences (i.e. 20,000 for every search), an independent search was conducted for the virus taxon (taxon ID: 10239). The search fetched 19,998 sequences with a match to RRARS. We also searched the 'Virus Pathogen' database [28], which is a depository of sequences exclusive to viruses.

Filtering of sequences and estimation of amino acid frequency: All FASTA sequences collected by the above method were then run through a python script to cross-check for the RRARS motif, and to identify the residue immediately before the RRARS motif (Putative P5 residue for Furin like enzymes). We further filtered the sequences to remove unknown, unnamed, hypothetical, predicted, and low-quality proteins. ,To prevent excessive repeats of the motif and skewing of the frequency of the residue at the P5 position, the script included a code to filter the isoforms of the same protein and only one isoform of a protein was allowed to retain. The resulting Excel file was sorted so as to classify proteins under different

organisms of origin (**Supplemental Table 1**). A frequency distribution graph was created to better visualize the number of times certain residues occurred at the putative P5 position (**Figure 1**). We found that residues R and A are the most frequent. One could consider those proteins with the R at 'P5' as substrates of Furin and Furin like enzymes. The sequences retrieved from the virus taxon were counted for the P5 residues (**Supplemental Table 1**). An independent serach was carried out on the ViPr pathogen Data base [28] (**Supplemental Table 2).** We did not look for P5 proline in the results from the Viral Pathogen database.

Construction of the Phylogenetic Tree

First, the CoV2 sequence of spike glycoprotein was queried from NCBI (ID: QHD43416). From the conserved domains via Pfam, the sequence was found to belong to a 'Corona_S2' superfamily (PF01601). Corona S2 family is a part of the fusion glycoprotein clan (CL0595) and interacts with the Fusion glycoprotein F0 family (PF00523). InterPro was searched for information on the Corona S2 glycoprotein family (IPR002552). By analysing the collection of proteins related to the family, 5188 (reviewed + unreviewed) Spike proteins were retrieved from UniProt using InterPro. Out of these 5188 proteins, 50 proteins were found to have undergone manual curation (SwissProt) and were included with the CoV2 Spike sequence. Six Feline coronaviruses (FCoV) isolates reported in [29] were fetched and added to the 50 curated spike proteins from Uniprot. A phylogenetic tree was built via Multiple Sequence Alignment (MSA) using the Clustal Omega service [30]. The phylogenetic tree thus obtained from the MSA in Newick format, was visualized using the Phylogenetic tree (Newick) viewer (Tool: Tree viewer - Online visualization of phylogenetic trees (Newick) and alignments) [31] and is shown in (**Figure 7)**. As one can see the FCoV isolates form a separate clade at the top of the tree while the Feline Infectious Peritonitis Virus (FIPV) sequence seems to be closer to the Canine family. These sequences were not close to the SARS CoV2.

Sequence comparison of ACE2 receptor and the RBD domains between FIPV and SARS CoV2

The FASTA sequences of Furin were taken for feline and human from UniProt; (UniProt IDs: M3W594, P09958). Pairwise sequence alignment was performed using the Smith-Waterman algorithm, implemented in the 'EMBOSS Water' tool [30]. The result of the sequence comparison among the Furin sequences revealed that both the sequences share 96.5% identity, 97.2% similarity, with one gap in the alignment (**Supplemental File 1**).

The FASTA sequences for the respective ACE2 receptors, of feline and human, were taken from UniProt; UniProt ID: Q56H28 [ACE2_FELCA], Q9BYF1 [ACE2_HUMAN]. Pairwise sequence alignment was performed using the Smith-Waterman algorithm, implemented in the 'EMBOSS Water' tool [30]. The result of the sequence comparison among the ACE2 receptors revealed that both the sequences share 85.2% identity, 92.3% similarity, with no gaps in the alignment (**Supplemental File 2**).

The FASTA sequences for the respective Spike proteins (containing the RBD domains), of FIPV and SARS CoV2, were taken from UniProt; UniProt ID: P10033 [SPIKE_FIPV], P0DTC2 [SPIKE_SARS2]. Pairwise sequence alignment was performed using the Smith-Waterman algorithm, implemented in the 'EMBOSS Water' tool. The result of the sequence comparison among the Spike proteins revealed that both the sequences share 26.1% identity, 40.5% similarity, with 26.3% gaps in the alignment (**Supplemental File 3**).

The known interactions between human ACE2 and CoV2 RBD were taken from PDBsum (PDB ID: 6m0j) and corresponding pairwise aligned residues, from feline ACE2 and FIPV RBD sequences respectively were mapped onto the diagram (**Figure 6**).

The X-ray structure of Furin co-crystallized with a peptide (PDB: 1P8J) was used to generate the model of Furin – PRRARS, Furin-RRKRRS complex. Molecular Dynamics simulations were carried out for both the complexes with ff14SB [32] force field using the pmemd.CUDA module from AMBER18 [33]. Hydrogen atoms were added and the N-terminus C-terminus of the peptides were capped with the residue ACE and NH2. All the simulation systems were neutralized with appropriate numbers of counter ions. The neutralized system was solvated in an octahedral box with TIP3P [34] water molecules, leaving at least 10 Å between the solute atoms and the borders of the box. MD simulations were carried out for 100ns in triplicates using the standard protocol published earlier [35]. Binding energy calculations were carried out with the last 50ns of trajectory using the protocol published earlier [35]. Simulation trajectories were visualized using VMD [36] and figures were generated using Pymol [37].

DECLARATION

- Ethics approval and consent to participate – Not applicable
- Consent for publication—all authors give their consent and share responsibility for the science in the manuscript.
- Competing interests None
- Funding Not available- the manuscript is primarily a product of work by two smart interns who came for their thesis as partial fulfilment of the Bachelor's degree requirement. Corresponding author has no funds for this project.

- Authors' contributions AB and SP Equal Contribution. All data base eperiments performed by the AB and SP. Docking and MD simulation by SR mentored by CV. PV conceived, designed and directed the project. AB, SP, PV and RV analysed the data and wrote the manuscript.
- If required any data will be made available.
- Acknowledgement- We thank Mahalakshmi Harish for careful reading of the manuscript and input. We thank all authors who would have contributed to the field but were not cited here.

References

1. Lau, S.K.P., et al., *Possible Bat Origin of Severe Acute Respiratory Syndrome Coronavirus 2.* Emerg Infect Dis, 2020. **26**(7).
2. Zhou, H., et al., *A Novel Bat Coronavirus Closely Related to SARS-CoV-2 Contains Natural Insertions at the S1/S2 Cleavage Site of the Spike Protein.* Curr Biol, 2020. **30**(11): p. 2196-2203 e3.
3. Liu, P., et al., *Are pangolins the intermediate host of the 2019 novel coronavirus (SARS-CoV-2)?* PLoS Pathog, 2020. **16**(5): p. e1008421.
4. Yamada, Y. and D.X. Liu, *Proteolytic activation of the spike protein at a novel RRRR/S motif is implicated in furin-dependent entry, syncytium formation, and infectivity of coronavirus infectious bronchitis virus in cultured cells.* J Virol, 2009. **83**(17): p. 8744-58.
5. Bottcher, E., et al., *Proteolytic activation of influenza viruses by serine proteases TMPRSS2 and HAT from human airway epithelium.* J Virol, 2006. **80**(19): p. 9896-8.
6. Baron, J., et al., *Matriptase, HAT, and TMPRSS2 activate the hemagglutinin of H9N2 influenza A viruses.* J Virol, 2013. **87**(3): p. 1811-20.
7. Coutard, B., et al., *The spike glycoprotein of the new coronavirus 2019-nCoV contains a furin-like cleavage site absent in CoV of the same clade.* Antiviral Res, 2020. **176**: p. 104742.
8. Chen, W.H., P.J. Hotez, and M.E. Bottazzi, *Potential for developing a SARS-CoV receptor-binding domain (RBD) recombinant protein as a heterologous human vaccine against coronavirus infectious disease (COVID)-19.* Hum Vaccin Immunother, 2020: p. 1-4.
9. Tai, W., et al., *Characterization of the receptor-binding domain (RBD) of 2019 novel coronavirus: implication for development of RBD protein as a viral attachment inhibitor and vaccine.* Cell Mol Immunol, 2020.
10. Jaimes, J.A., J.K. Millet, and G.R. Whittaker, *Proteolytic Cleavage of the SARS-CoV-2 Spike Protein and the Role of the Novel S1/S2 Site.* iScience, 2020. **23**(6): p. 101212.
11. Hoffmann, M., H. Kleine-Weber, and S. Pohlmann, *A Multibasic Cleavage Site in the Spike Protein of SARS-CoV-2 Is Essential for Infection of Human Lung Cells.* Mol Cell, 2020.
12. Izaguirre, G., *The Proteolytic Regulation of Virus Cell Entry by Furin and Other Proprotein Convertases.* Viruses, 2019. **11**(9).
13. Barrett, A.J., N.D. Rawlings, and E.A. O'Brien, *The MEROPS database as a protease information system.* J Struct Biol, 2001. **134**(2-3): p. 95-102.
14. Licitra, B.N., et al., *Mutation in spike protein cleavage site and pathogenesis of feline coronavirus.* Emerg Infect Dis, 2013. **19**(7): p. 1066-73.
15. Shiryaev, S.A., et al., *High-resolution analysis and functional mapping of cleavage sites and substrate proteins of furin in the human proteome.* PLoS One, 2013. **8**(1): p. e54290.
16. Thomas, G., *Furin at the cutting edge: from protein traffic to embryogenesis and disease.* Nat Rev Mol Cell Biol, 2002. **3**(10): p. 753-66.
17. Hoffmann, M., et al., *SARS-CoV-2 Cell Entry Depends on ACE2 and TMPRSS2 and Is Blocked by a Clinically Proven Protease Inhibitor.* Cell, 2020. **181**(2): p. 271-280 e8.
18. Tse, L.V., et al., *A novel activation mechanism of avian influenza virus H9N2 by furin.* J Virol, 2014. **88**(3): p. 1673-83.

19. Venkatraman, P., et al., *A sequence and structure based method to predict putative substrates, functions and regulatory networks of endo proteases.* PLoS One, 2009. **4**(5): p. e5700.
20. Wadhawan, V., et al., *From prediction to experimental validation: desmoglein 2 is a functionally relevant substrate of matriptase in epithelial cells and their reciprocal relationship is important for cell adhesion.* Biochem J, 2012. **447**(1): p. 61-70.
21. Dhamne, H., A.G. Chande, and R. Mukhopadhyaya, *Lentiviral vector platform for improved erythropoietin expression concomitant with shRNA mediated host cell elastase down regulation.* Plasmid, 2014. **71**: p. 1-7.
22. Rolas, L., et al., *NADPH oxidase depletion in neutrophils from patients with cirrhosis and restoration via toll-like receptor 7/8 activation.* Gut, 2018. **67**(8): p. 1505-1516.
23. Wahba, L., et al., *An Extensive Meta-Metagenomic Search Identifies SARS-CoV-2-Homologous Sequences in Pangolin Lung Viromes.* mSphere, 2020. **5**(3).
24. Shi, J., et al., *Susceptibility of ferrets, cats, dogs, and other domesticated animals to SARS-coronavirus 2.* Science, 2020.
25. Andersen, K.G., et al., *The proximal origin of SARS-CoV-2.* Nat Med, 2020. **26**(4): p. 450-452.
26. Beaulieu, A., et al., *Matriptase proteolytically activates influenza virus and promotes multicycle replication in the human airway epithelium.* J Virol, 2013. **87**(8): p. 4237-51.
27. Belouzard, S., V.C. Chu, and G.R. Whittaker, *Activation of the SARS coronavirus spike protein via sequential proteolytic cleavage at two distinct sites.* Proc Natl Acad Sci U S A, 2009. **106**(14): p. 5871-6.
28. Pickett, B.E., et al., *ViPR: an open bioinformatics database and analysis resource for virology research.* Nucleic Acids Res, 2012. **40**(Database issue): p. D593-8.
29. Lewis, C.S., et al., *Genotyping coronaviruses associated with feline infectious peritonitis.* J Gen Virol, 2015. **96**(Pt 6): p. 1358-1368.
30. Madeira, F., et al., *The EMBL-EBI search and sequence analysis tools APIs in 2019.* Nucleic Acids Res, 2019. **47**(W1): p. W636-W641.
31. Huerta-Cepas, J., F. Serra, and P. Bork, *ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data.* Mol Biol Evol, 2016. **33**(6): p. 1635-8.
32. Maier, J.A., et al., *ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB.* J Chem Theory Comput, 2015. **11**(8): p. 3696-713.
33. Case, D.A.e.a., *AMBER 18.* University of California, San Francisco, 2018.
34. Jorgensen, W.L., et al., *Comparison of simple potential functions for simulating liquid water.* The Journal of Chemical Physics, 1983. **79**(2): p. 926-935.
35. Kannan, S., et al., *Inhibiting S100B(ββ) for Activating Wild-Type p53: Design of Stapled Peptides.* ACS Omega, 2019. **4**(3): p. 5335-5344.
36. Humphrey, W., A. Dalke, and K. Schulten, *VMD: Visual molecular dynamics.* Journal of Molecular Graphics, 1996. **14**(1): p. 33-38.
37. DeLano, W.L., *The PyMOL molecular graphics system.* San Carlos CA, USA, 2002. **De Lano Scientific**.

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
BtKYNL63-15	NL63-related b	M protein	1.0	111-115	RRARS
28O	Alphacoronavi	surface glycop	1.0	788-792	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
00021_12	Hepatitis C vir	NS5B RNA-de	1.0	258-262	RRARS
00021_12	Hepatitis C vir	NS5B protein	1.0	258-262	RRARS
SA14-14-2	Japanese enc	RNA-depende	1.0	41-45	RRARS
SA14-14-2	Japanese enc	polyprotein	1.0	2568-2572	RRARS

Strain Nam	Species Na	Protein Nar	Score	Range	Matched Sequence
LN3131-1	Beluga wha	protein Mo	1.0	87-91	RRARS
UNKNOWN	Suid alphal	hypothetica	1.0	193-197	RRARS
UNKNOWN	Suid alphal	hypothetica	1.0	267-271	RRARS
UNKNOWN	Suid alphal	hypothetica	1.0	193-197	RRARS
UNKNOWN	Suid alphal	hypothetica	1.0	267-271	RRARS
17	Human alp	large tegun	1.0	2661-2665	RRARS
MV 5-4	Saimiriine a	myristylate	1.0	68-72	RRARS
132/1998	Human alp	large tegun	1.0	2656-2660	RRARS
160/1982	Human alp	large tegun	1.0	2656-2660	RRARS
66/2007	Human alp	large tegun	1.0	2656-2660	RRARS
1394/2005	Human alp	large tegun	1.0	2656-2660	RRARS
270/2007	Human alp	large tegun	1.0	2656-2660	RRARS
1319/2005	Human alp	large tegun	1.0	2656-2660	RRARS
369/2007	Human alp	large tegun	1.0	2656-2660	RRARS
3083/2008	Human alp	large tegun	1.0	2656-2660	RRARS
172/2010	Human alp	large tegun	1.0	2656-2660	RRARS
2158/2007	Human alp	large tegun	1.0	2656-2660	RRARS
2C	Human alp	Deneddyla	1.0	1226-1230	RRARS
McKrae	Human alp	large tegun	1.0	2656-2660	RRARS
MacIntyre	Human alp	large tegun	1.0	2656-2660	RRARS
HSV-hepat	Human alp	UL36	1.0	2651-2655	RRARS
2018-5967	Human alp	UL36	1.0	2656-2660	RRARS
2018-5965	Human alp	UL36	1.0	2646-2650	RRARS
2018-5968	Human alp	UL36	1.0	2646-2650	RRARS
2018-5971	Human alp	UL36	1.0	2646-2650	RRARS
2018-5969	Human alp	UL36	1.0	2646-2650	RRARS
2018-5973	Human alp	UL36	1.0	2646-2650	RRARS
2018-5970	Human alp	UL36	1.0	2646-2650	RRARS
2018-5972	Human alp	UL36	1.0	2646-2650	RRARS
2000-3429	Human alp	hypothetica	1.0	76-80	RRARS
2010-8179	Human alp	hypothetica	1.0	76-80	RRARS
2000-9815	Human alp	hypothetica	1.0	76-80	RRARS
2007-2205	Human alp	UL46	1.0	3-7	RRARS
HSV-H141	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H151	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H131	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H131	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H121	Human alp	UL36	1.0	2682-2686	RRARS
HSV-H121	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H121	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H121	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H121	Human alp	UL36	1.0	2656-2660	RRARS
HSV-H121	Human alp	UL36	1.0	2656-2660	RRARS
Ty148	Human alp	UL36	1.0	2656-2660	RRARS
Ty25	Human alp	UL36	1.0	2659-2663	RRARS
K86	Human alp	UL36	1.0	2659-2663	RRARS
17	Human her	very large t	1.0	2686-2690	RRARS
RH2	Human alp	large tegun	1.0	2656-2660	RRARS
HSV-v29_s	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_s	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_s	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_s	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_s	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_d	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_d	Human alp	UL36	1.0	2656-2660	RRARS

HSV-v29_d	Human alp	UL36	1.0	2656-2660	RRARS
HSV-v29_s	Human alp	UL36	1.0	2656-2660	RRARS
2011-1026	Human alp	UL36	1.0	2651-2655	RRARS
2014-3233	Human alp	UL36	1.0	2282-2286	RRARS
2011-1274	Human alp	UL36	1.0	2651-2655	RRARS
2016-1040	Human alp	UL36	1.0	2651-2655	RRARS
2006-5768	Human alp	UL36	1.0	2656-2660	RRARS
2007-1642	Human alp	UL36	1.0	2686-2690	RRARS
2009-2037	Human alp	UL36	1.0	2686-2690	RRARS
2003-1575	Human alp	hypothetica	1.0	47-51	RRARS
2010-3208	Human alp	UL36	1.0	2661-2665	RRARS
2011-5409	Human alp	UL36	1.0	2646-2650	RRARS
2011-1271	Human alp	UL36	1.0	2646-2650	RRARS
2012-3580	Human alp	UL36	1.0	2681-2685	RRARS
1995-6317	Human alp	UL36	1.0	2646-2650	RRARS
1998-7487	Human alp	UL36	1.0	2646-2650	RRARS
2011-1524	Human alp	UL36	1.0	2681-2685	RRARS
2011-1631	Human alp	hypothetica	1.0	54-58	RRARS
2010-2868	Human alp	UL36	1.0	2656-2660	RRARS
2011-1271	Human alp	UL36	1.0	2646-2650	RRARS
2000-2419	Human alp	UL36	1.0	2646-2650	RRARS
1997-5801	Human alp	UL36	1.0	2646-2650	RRARS
2009-2557	Human alp	hypothetica	1.0	47-51	RRARS
2009-2996	Human alp	UL36	1.0	2649-2653	RRARS
2006-5763	Human alp	hypothetica	1.0	54-58	RRARS
2011-3116	Human alp	UL36	1.0	2646-2650	RRARS
HSV1-CUL	Human alp	UL36_1	1.0	2282-2286	RRARS
HSV1-CUL	Human alp	UL36	1.0	2656-2660	RRARS
HSV1-CUL	Human alp	UL36	1.0	2681-2685	RRARS
HSV1-CUL	Human alp	UL36_1	1.0	2282-2286	RRARS
HSV1-CUL	Human alp	UL36	1.0	2685-2689	RRARS
HSV1-CUL	Human alp	UL36	1.0	2646-2650	RRARS
HSV1-CUL	Human alp	UL36	1.0	2681-2685	RRARS
HSV1-CUL	Human alp	UL36	1.0	2646-2650	RRARS
HSV1-CUL	Human alp	UL36_1	1.0	2282-2286	RRARS
HSV-N-7	Human alp	UL36	1.0	2656-2660	RRARS
HSV-R-13	Human alp	UL36	1.0	2656-2660	RRARS
K(delta)25/	Human alp	large tegun	1.0	2656-2660	RRARS
KQ	Macacine a	large tegun	1.0	1685-1689	RRARS
1504-11	Macacine a	large tegun	1.0	1685-1689	RRARS
ZW6	Human alp	large tegun	1.0	2656-2660	RRARS
SC16	Human alp	large tegun	1.0	2656-2660	RRARS
914-R2	Human her	large tegun	1.0	2656-2660	RRARS
20-14-2	Human her	large tegun	1.0	2656-2660	RRARS
20-14-23	Human her	large tegun	1.0	2656-2660	RRARS
914-B	Human her	large tegun	1.0	2656-2660	RRARS
914-T2	Human her	large tegun	1.0	2656-2660	RRARS
20-14-1	Human her	large tegun	1.0	2656-2660	RRARS
914-A3	Human her	large tegun	1.0	2656-2660	RRARS
20-14-22	Human her	large tegun	1.0	2656-2660	RRARS
20-14-7	Human her	large tegun	1.0	2661-2665	RRARS
914-E3	Human her	large tegun	1.0	2656-2660	RRARS
914-Z	Human her	large tegun	1.0	2656-2660	RRARS
914-Y2	Human her	large tegun	1.0	2656-2660	RRARS
IV-1	Human her	large tegun	1.0	2656-2660	RRARS
20-14-24	Human her	large tegun	1.0	2656-2660	RRARS

IV-6	Human her	large tegun	1.0	2656-2660 RRARS
914-H2	Human her	large tegun	1.0	2656-2660 RRARS
20-14-18	Human her	large tegun	1.0	2656-2660 RRARS
914-N3	Human her	large tegun	1.0	2656-2660 RRARS
20-14-8	Human her	large tegun	1.0	2661-2665 RRARS
IV-7	Human her	large tegun	1.0	2656-2660 RRARS
914-D2	Human her	large tegun	1.0	2656-2660 RRARS
914-B2	Human her	large tegun	1.0	2656-2660 RRARS
914-Q	Human her	large tegun	1.0	2656-2660 RRARS
IV-2	Human her	large tegun	1.0	2656-2660 RRARS
914-O2	Human her	large tegun	1.0	2656-2660 RRARS
OD4	Human her	large tegun	1.0	2661-2665 RRARS
B^3x1.2	Human her	large tegun	1.0	1462-1466 RRARS
B^3x1.1	Human her	large tegun	1.0	1462-1466 RRARS
KOS 1.1	Human her	large tegun	1.0	2656-2660 RRARS
KOS	Human her	large tegun	1.0	2656-2660 RRARS
KOS63	Human her	UL36	1.0	2656-2660 RRARS
KOS79	Human her	UL36	1.0	2656-2660 RRARS
RDH193	Human her	UL36	1.0	2656-2660 RRARS
McKrae	Human her	UL36	1.0	2656-2660 RRARS
81L	Human her	large tegun	1.0	2661-2665 RRARS
78S	Human her	large tegun	1.0	2661-2665 RRARS
65M	Human her	large tegun	1.0	2661-2665 RRARS
5-4-2	Human her	large tegun	1.0	2661-2665 RRARS
83M	Human her	large tegun	1.0	2661-2665 RRARS
26S	Human her	large tegun	1.0	2661-2665 RRARS
82S	Human her	large tegun	1.0	2661-2665 RRARS
10-6-2	Human her	large tegun	1.0	2661-2665 RRARS
47M	Human her	large tegun	1.0	2661-2665 RRARS
31XL	Human her	large tegun	1.0	2661-2665 RRARS
16S	Human her	large tegun	1.0	2661-2665 RRARS
10-1-2	Human her	large tegun	1.0	2661-2665 RRARS
10-5-1	Human her	large tegun	1.0	2661-2665 RRARS
76M	Human her	large tegun	1.0	2661-2665 RRARS
5-1-1	Human her	large tegun	1.0	2661-2665 RRARS
12-12-2	Human her	large tegun	1.0	2661-2665 RRARS
27S	Human her	large tegun	1.0	2661-2665 RRARS
10-6-1	Human her	large tegun	1.0	2661-2665 RRARS
5-5-2	Human her	large tegun	1.0	2661-2665 RRARS
11M	Human her	large tegun	1.0	2661-2665 RRARS
19Lsyn	Human her	large tegun	1.0	2661-2665 RRARS
2-5-3	Human her	large tegun	1.0	2661-2665 RRARS
10-14-1	Human her	large tegun	1.0	2661-2665 RRARS
10-7-1	Human her	large tegun	1.0	2661-2665 RRARS
20L	Human her	large tegun	1.0	2661-2665 RRARS
2-4-2	Human her	large tegun	1.0	2661-2665 RRARS
10-11-2	Human her	large tegun	1.0	2661-2665 RRARS
12-12-67	Human her	large tegun	1.0	2661-2665 RRARS
5-2-1	Human her	large tegun	1.0	2661-2665 RRARS
10-6-3	Human her	large tegun	1.0	2661-2665 RRARS
CJ994	Human her	large tegun	1.0	2661-2665 RRARS
3M	Human her	large tegun	1.0	2661-2665 RRARS
66S	Human her	large tegun	1.0	2661-2665 RRARS
8S	Human her	large tegun	1.0	2661-2665 RRARS
36L	Human her	large tegun	1.0	2661-2665 RRARS
4M	Human her	large tegun	1.0	2661-2665 RRARS

10-2-2	Human her	large tegun	1.0	2661-2665	RRARS
57M	Human her	large tegun	1.0	2661-2665	RRARS
34L	Human her	large tegun	1.0	2661-2665	RRARS
10-2-3	Human her	large tegun	1.0	2661-2665	RRARS
HSV-1/011	Human her	large tegun	1.0	2656-2660	RRARS
H166syn	Human alp	UL36	1.0	2656-2660	RRARS
H166	Human alp	UL36	1.0	2656-2660	RRARS
F	Human alp	UL36	1.0	2656-2660	RRARS
F	Human alp	UL36	1.0	2656-2660	RRARS
KOS	Human alp	UL36	1.0	2656-2660	RRARS
KOS	Human alp	UL36	1.0	2656-2660	RRARS
KOS	Human alp	UL36	1.0	2656-2660	RRARS
MacIntyre	Human her	UL36	1.0	2656-2660	RRARS
E90-136	Macacine a	envelope g	1.0	361-365	RRARS
OU1-76	Papiine her	DNA polym	1.0	590-594	RRARS
RE	Human her	large tegun	1.0	2656-2660	RRARS
McKrae	Human her	large tegun	1.0	2661-2665	RRARS
KOS	Human her	large tegun	1.0	2656-2660	RRARS
KOS	Human her	large tegun	1.0	2656-2660	RRARS
CJ360	Human her	UL36	1.0	2661-2665	RRARS
134	Human her	UL36	1.0	2661-2665	RRARS
17	Human her	large tegun	1.0	2661-2665	RRARS
MV 5-4	Saimiriine h	myristylate	1.0	68-72	RRARS
R62	Human her	large tegun	1.0	2659-2663	RRARS
R11	Human her	large tegun	1.0	2656-2660	RRARS
S25	Human her	large tegun	1.0	2659-2663	RRARS
S23	Human her	large tegun	1.0	2659-2663	RRARS
E19	Human her	large tegun	1.0	2656-2660	RRARS
E14	Human her	large tegun	1.0	2656-2660	RRARS
E03	Human her	large tegun	1.0	2656-2660	RRARS
CR38	Human her	large tegun	1.0	2656-2660	RRARS
E35	Human her	large tegun	1.0	2656-2660	RRARS
E25	Human her	large tegun	1.0	2656-2660	RRARS
E23	Human her	large tegun	1.0	2656-2660	RRARS
E22	Human her	large tegun	1.0	2656-2660	RRARS
E15	Human her	large tegun	1.0	2656-2660	RRARS
E13	Human her	large tegun	1.0	2656-2660	RRARS
E12	Human her	large tegun	1.0	2656-2660	RRARS
E11	Human her	large tegun	1.0	2656-2660	RRARS
E10	Human her	large tegun	1.0	2656-2660	RRARS
E08	Human her	large tegun	1.0	2656-2660	RRARS
E07	Human her	large tegun	1.0	2656-2660	RRARS
E06	Human her	large tegun	1.0	2656-2660	RRARS
H129	Human her	large tegun	1.0	2656-2660	RRARS
F	Human her	large tegun	1.0	2656-2660	RRARS
17	Human her	large tegun	1.0	2661-2665	RRARS
HF	Human her	UL36	1.0	2681-2685	RRARS
UNKNOWN	Chelonid h	F-US11 pro	1.0	2-6	RRARS
UNKNOWN	Cricetid ga	hypothetica	1.0	123-127	RRARS
UNKNOWN	Cricetid he	hypothetica	1.0	123-127	RRARS
UNKNOWN	Cricetid he	hypothetica	1.0	123-127	RRARS
86/67	Equid gam	protein E6(	1.0	85-89	RRARS
G9/92	Equid herp	protein E6(	1.0	85-89	RRARS
86/67	Equid herp	protein E6(	1.0	85-89	RRARS
England	Murid beta	E86	1.0	1078-1082	RRARS
G3e	Murid herp	m155 prote	1.0	299-303	RRARS

K7	Murid herp	m155 prote	1.0	299-303	RRARS
s02_sk267	Murid betal	m155 prote	1.0	299-303	RRARS
SN51_s45_	Murid betal	m155 prote	1.0	298-302	RRARS
s88_sk273	Murid betal	m155 prote	1.0	298-302	RRARS
ALL-03	Unclassifie	a86	1.0	1078-1082	RRARS
Berlin	Murid herp	B86	1.0	1078-1082	RRARS
England	Murid betal	E86	1.0	1078-1082	RRARS
UNKNOWN	Human gar	protein RPI	1.0	61-65	RRARS
UNKNOWN	Human gar	nuclear ant	1.0	243-247	RRARS
UNKNOWN	Human gar	virion prote	1.0	144-148	RRARS
LCL8664	Macacine g	BLRF2	1.0	145-149	RRARS
AG876	Human gar	EBNA-3B	1.0	243-247	RRARS
AG876	Human gar	BLRF2	1.0	144-148	RRARS
NOSB	Human gar	BLRF2	1.0	144-148	RRARS
PTLB3	Human gar	BLRF2	1.0	144-148	RRARS
PTBL2	Human gar	BLRF2	1.0	144-148	RRARS
PTBL1	Human gar	BLRF2	1.0	144-148	RRARS
NKTL2	Human gar	BLRF2	1.0	144-148	RRARS
DLBCL4	Human gar	BLRF2	1.0	144-148	RRARS
DLBCL2	Human gar	BLRF2	1.0	141-145	RRARS
CTCL1	Human gar	BLRF2	1.0	141-145	RRARS
ARL2	Human gar	BLRF2	1.0	144-148	RRARS
AIL16	Human gar	BLRF2	1.0	141-145	RRARS
AIL15	Human gar	BLRF2	1.0	144-148	RRARS
AIL14	Human gar	BLRF2	1.0	144-148	RRARS
AIL13	Human gar	BLRF2	1.0	141-145	RRARS
AIL7	Human gar	BLRF2	1.0	144-148	RRARS
AIL2	Human gar	BLRF2	1.0	144-148	RRARS
AIL1	Human gar	BLRF2	1.0	144-148	RRARS
NPCT115	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT115	Human gar	virion prote	1.0	144-148	RRARS
NPCT114	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT114	Human gar	virion prote	1.0	144-148	RRARS
NPCT113	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT113	Human gar	virion prote	1.0	144-148	RRARS
NPCT112	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT112	Human gar	virion prote	1.0	144-148	RRARS
NPCT111	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT111	Human gar	virion prote	1.0	144-148	RRARS
NPCT110	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT110	Human gar	virion prote	1.0	144-148	RRARS
NPCT109	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT109	Human gar	virion prote	1.0	144-148	RRARS
NPCT108	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT108	Human gar	virion prote	1.0	144-148	RRARS
NPCT107	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT107	Human gar	virion prote	1.0	144-148	RRARS
NPCT106	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT106	Human gar	virion prote	1.0	144-148	RRARS
NPCT105	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT105	Human gar	virion prote	1.0	144-148	RRARS
NPCT104	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT104	Human gar	virion prote	1.0	144-148	RRARS
NPCT103	Human gar	nuclear ant	1.0	243-247	RRARS
NPCT103	Human gar	virion prote	1.0	144-148	RRARS
NPCT102	Human gar	nuclear ant	1.0	243-247	RRARS

NPCT102	Human gar	virion prote	1.0	144-148 RRARS
NPCT101	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT101	Human gar	virion prote	1.0	144-148 RRARS
NPCT100	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT100	Human gar	virion prote	1.0	144-148 RRARS
NPCT099	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT099	Human gar	virion prote	1.0	144-148 RRARS
NPCT098	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT098	Human gar	virion prote	1.0	144-148 RRARS
NPCT096	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT096	Human gar	virion prote	1.0	144-148 RRARS
NPCT094	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT094	Human gar	virion prote	1.0	144-148 RRARS
NPCT093	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT093	Human gar	virion prote	1.0	144-148 RRARS
NPCT092	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT092	Human gar	virion prote	1.0	144-148 RRARS
NPCT091	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT091	Human gar	virion prote	1.0	144-148 RRARS
NPCT090	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT090	Human gar	virion prote	1.0	144-148 RRARS
NPCT089	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT089	Human gar	virion prote	1.0	144-148 RRARS
NPCT088	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT088	Human gar	virion prote	1.0	144-148 RRARS
NPCT087	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT087	Human gar	virion prote	1.0	144-148 RRARS
NPCT086	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT086	Human gar	virion prote	1.0	144-148 RRARS
NPCT085	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT085	Human gar	virion prote	1.0	144-148 RRARS
NPCT084	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT084	Human gar	virion prote	1.0	144-148 RRARS
NPCT083	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT083	Human gar	virion prote	1.0	144-148 RRARS
NPCT082	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT082	Human gar	virion prote	1.0	144-148 RRARS
NPCT081	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT081	Human gar	virion prote	1.0	144-148 RRARS
NPCT080	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT080	Human gar	virion prote	1.0	144-148 RRARS
NPCT078	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT078	Human gar	virion prote	1.0	144-148 RRARS
NPCT077	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT077	Human gar	virion prote	1.0	144-148 RRARS
NPCT076	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT076	Human gar	virion prote	1.0	144-148 RRARS
NPCT075	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT075	Human gar	virion prote	1.0	144-148 RRARS
NPCT074S	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT074S	Human gar	virion prote	1.0	144-148 RRARS
NPCT074	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT074	Human gar	virion prote	1.0	144-148 RRARS
NPCT073	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT073	Human gar	virion prote	1.0	144-148 RRARS
NPCT072	Human gar	nuclear ant	1.0	243-247 RRARS

NPCT072	Human gar	virion prote	1.0	144-148 RRARS
NPCT071	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT071	Human gar	virion prote	1.0	144-148 RRARS
NPCT070	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT070	Human gar	virion prote	1.0	144-148 RRARS
NPCT069	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT069	Human gar	virion prote	1.0	144-148 RRARS
NPCT068	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT068	Human gar	virion prote	1.0	144-148 RRARS
NPCT067	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT067	Human gar	virion prote	1.0	144-148 RRARS
NPCT066	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT066	Human gar	virion prote	1.0	144-148 RRARS
NPCT065	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT065	Human gar	virion prote	1.0	144-148 RRARS
NPCT064	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT064	Human gar	virion prote	1.0	144-148 RRARS
NPCT063	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT063	Human gar	virion prote	1.0	144-148 RRARS
NPCT062	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT062	Human gar	virion prote	1.0	144-148 RRARS
NPCT061	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT061	Human gar	virion prote	1.0	144-148 RRARS
NPCT060	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT060	Human gar	virion prote	1.0	144-148 RRARS
NPCT059	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT059	Human gar	virion prote	1.0	144-148 RRARS
NPCT058M	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT058M	Human gar	virion prote	1.0	144-148 RRARS
NPCT058	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT058	Human gar	virion prote	1.0	144-148 RRARS
NPCT057M	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT057M	Human gar	virion prote	1.0	144-148 RRARS
NPCT057	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT057	Human gar	virion prote	1.0	144-148 RRARS
NPCT056M	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT056M	Human gar	virion prote	1.0	144-148 RRARS
NPCT056	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT056	Human gar	virion prote	1.0	144-148 RRARS
NPCT055M	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT055M	Human gar	virion prote	1.0	144-148 RRARS
NPCT055	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT055	Human gar	virion prote	1.0	144-148 RRARS
NPCT054M	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT054M	Human gar	virion prote	1.0	144-148 RRARS
NPCT054	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT054	Human gar	virion prote	1.0	144-148 RRARS
NPCT053	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT053	Human gar	virion prote	1.0	144-148 RRARS
NPCT052	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT052	Human gar	virion prote	1.0	144-148 RRARS
NPCT050	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT050	Human gar	virion prote	1.0	144-148 RRARS
NPCT049	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT049	Human gar	virion prote	1.0	144-148 RRARS
NPCT048	Human gar	nuclear ant	1.0	243-247 RRARS

NPCT048	Human gar	virion prote	1.0	144-148 RRARS
NPCT047	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT047	Human gar	virion prote	1.0	144-148 RRARS
NPCT046	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT046	Human gar	virion prote	1.0	144-148 RRARS
NPCT045	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT045	Human gar	virion prote	1.0	144-148 RRARS
NPCT043	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT043	Human gar	virion prote	1.0	144-148 RRARS
NPCT042	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT042	Human gar	virion prote	1.0	144-148 RRARS
NPCT041	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT041	Human gar	virion prote	1.0	144-148 RRARS
NPCT040	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT040	Human gar	virion prote	1.0	144-148 RRARS
NPCT039	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT039	Human gar	virion prote	1.0	144-148 RRARS
NPCT038	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT038	Human gar	virion prote	1.0	144-148 RRARS
NPCT037	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT037	Human gar	virion prote	1.0	144-148 RRARS
NPCT036	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT036	Human gar	virion prote	1.0	144-148 RRARS
NPCT035	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT035	Human gar	virion prote	1.0	144-148 RRARS
NPCT033	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT033	Human gar	virion prote	1.0	144-148 RRARS
NPCT032	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT032	Human gar	virion prote	1.0	144-148 RRARS
NPCT031	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT031	Human gar	virion prote	1.0	144-148 RRARS
NPCT029	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT029	Human gar	virion prote	1.0	144-148 RRARS
NPCT028-2	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT028-2	Human gar	virion prote	1.0	144-148 RRARS
NPCT027	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT027	Human gar	virion prote	1.0	144-148 RRARS
NPCT025	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT025	Human gar	virion prote	1.0	144-148 RRARS
NPCT024	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT024	Human gar	virion prote	1.0	144-148 RRARS
NPCT023	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT023	Human gar	virion prote	1.0	144-148 RRARS
NPCT022	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT022	Human gar	virion prote	1.0	144-148 RRARS
NPCT021	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT021	Human gar	virion prote	1.0	144-148 RRARS
NPCT020-2	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT020-2	Human gar	virion prote	1.0	144-148 RRARS
NPCT019	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT019	Human gar	virion prote	1.0	144-148 RRARS
NPCT018	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT018	Human gar	virion prote	1.0	144-148 RRARS
NPCT017	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT017	Human gar	virion prote	1.0	144-148 RRARS
NPCT015	Human gar	nuclear ant	1.0	243-247 RRARS

NPCT015	Human gar	virion prote	1.0	144-148 RRARS
NPCT014	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT014	Human gar	virion prote	1.0	144-148 RRARS
NPCT013	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT013	Human gar	virion prote	1.0	144-148 RRARS
NPCT012	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT012	Human gar	virion prote	1.0	144-148 RRARS
NPCT011	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT011	Human gar	virion prote	1.0	144-148 RRARS
NPCT010	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT010	Human gar	virion prote	1.0	144-148 RRARS
NPCT009	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT009	Human gar	virion prote	1.0	144-148 RRARS
NPCT008	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT008	Human gar	virion prote	1.0	144-148 RRARS
NPCT007	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT007	Human gar	virion prote	1.0	144-148 RRARS
NPCT006	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT006	Human gar	virion prote	1.0	144-148 RRARS
NPCT005	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT005	Human gar	virion prote	1.0	144-148 RRARS
NPCT004	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT004	Human gar	virion prote	1.0	144-148 RRARS
NPCT003	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT003	Human gar	virion prote	1.0	144-148 RRARS
NPCT002	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT002	Human gar	virion prote	1.0	144-148 RRARS
NPCT001	Human gar	nuclear ant	1.0	243-247 RRARS
NPCT001	Human gar	virion prote	1.0	144-148 RRARS
NPCS054	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS054	Human gar	virion prote	1.0	144-148 RRARS
NPCS052	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS052	Human gar	virion prote	1.0	144-148 RRARS
NPCS051	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS051	Human gar	virion prote	1.0	144-148 RRARS
NPCS050	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS050	Human gar	virion prote	1.0	144-148 RRARS
NPCS049	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS049	Human gar	virion prote	1.0	144-148 RRARS
NPCS048	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS048	Human gar	virion prote	1.0	144-148 RRARS
NPCS047	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS047	Human gar	virion prote	1.0	144-148 RRARS
NPCS046	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS046	Human gar	virion prote	1.0	144-148 RRARS
NPCS045	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS045	Human gar	virion prote	1.0	144-148 RRARS
NPCS044	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS044	Human gar	virion prote	1.0	144-148 RRARS
NPCS042	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS042	Human gar	virion prote	1.0	144-148 RRARS
NPCS040	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS040	Human gar	virion prote	1.0	144-148 RRARS
NPCS039	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS039	Human gar	virion prote	1.0	144-148 RRARS
NPCS038	Human gar	nuclear ant	1.0	243-247 RRARS

NPCS038	Human gar	virion prote	1.0	144-148 RRARS
NPCS035	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS035	Human gar	virion prote	1.0	144-148 RRARS
NPCS034	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS034	Human gar	virion prote	1.0	144-148 RRARS
NPCS033	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS033	Human gar	virion prote	1.0	144-148 RRARS
NPCS031	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS031	Human gar	virion prote	1.0	144-148 RRARS
NPCS030	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS030	Human gar	virion prote	1.0	144-148 RRARS
NPCS029	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS029	Human gar	virion prote	1.0	144-148 RRARS
NPCS028	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS028	Human gar	virion prote	1.0	144-148 RRARS
NPCS027	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS027	Human gar	virion prote	1.0	144-148 RRARS
NPCS026	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS026	Human gar	virion prote	1.0	144-148 RRARS
NPCS025	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS025	Human gar	virion prote	1.0	144-148 RRARS
NPCS024	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS024	Human gar	virion prote	1.0	144-148 RRARS
NPCS023	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS023	Human gar	virion prote	1.0	144-148 RRARS
NPCS022	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS022	Human gar	virion prote	1.0	144-148 RRARS
NPCS021	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS021	Human gar	virion prote	1.0	144-148 RRARS
NPCS019	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS019	Human gar	virion prote	1.0	144-148 RRARS
NPCS018	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS018	Human gar	virion prote	1.0	144-148 RRARS
NPCS017	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS017	Human gar	virion prote	1.0	144-148 RRARS
NPCS016	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS016	Human gar	virion prote	1.0	144-148 RRARS
NPCS014	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS014	Human gar	virion prote	1.0	144-148 RRARS
NPCS013	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS013	Human gar	virion prote	1.0	144-148 RRARS
NPCS012	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS012	Human gar	virion prote	1.0	144-148 RRARS
NPCS011	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS011	Human gar	virion prote	1.0	144-148 RRARS
NPCS010	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS010	Human gar	virion prote	1.0	144-148 RRARS
NPCS009	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS009	Human gar	virion prote	1.0	144-148 RRARS
NPCS008	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS008	Human gar	virion prote	1.0	144-148 RRARS
NPCS007	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS007	Human gar	virion prote	1.0	144-148 RRARS
NPCS006	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS006	Human gar	virion prote	1.0	144-148 RRARS
NPCS005	Human gar	nuclear ant	1.0	243-247 RRARS

NPCS005	Human gar	virion prote	1.0	144-148 RRARS
NPCS003-2	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS003-2	Human gar	virion prote	1.0	144-148 RRARS
NPCS002	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS002	Human gar	virion prote	1.0	144-148 RRARS
NPCS001	Human gar	nuclear ant	1.0	243-247 RRARS
NPCS001	Human gar	virion prote	1.0	144-148 RRARS
NPCP001	Human gar	nuclear ant	1.0	243-247 RRARS
NPCP001	Human gar	virion prote	1.0	144-148 RRARS
NNPCT005	Human gar	nuclear ant	1.0	243-247 RRARS
NNPCT005	Human gar	virion prote	1.0	144-148 RRARS
NNPCT004	Human gar	nuclear ant	1.0	243-247 RRARS
NNPCT004	Human gar	virion prote	1.0	144-148 RRARS
NNPCT003	Human gar	nuclear ant	1.0	243-247 RRARS
NNPCT003	Human gar	virion prote	1.0	144-148 RRARS
NNPCT002	Human gar	nuclear ant	1.0	243-247 RRARS
NNPCT002	Human gar	virion prote	1.0	144-148 RRARS
NNPCT001	Human gar	nuclear ant	1.0	243-247 RRARS
NNPCT001	Human gar	virion prote	1.0	144-148 RRARS
NKLT007	Human gar	nuclear ant	1.0	243-247 RRARS
NKLT007	Human gar	virion prote	1.0	144-148 RRARS
NKLT006	Human gar	nuclear ant	1.0	243-247 RRARS
NKLT006	Human gar	virion prote	1.0	144-148 RRARS
NKLT004	Human gar	nuclear ant	1.0	243-247 RRARS
NKLT004	Human gar	virion prote	1.0	144-148 RRARS
NKLT003-2	Human gar	nuclear ant	1.0	243-247 RRARS
NKLT003-2	Human gar	virion prote	1.0	144-148 RRARS
NKLT002	Human gar	nuclear ant	1.0	243-247 RRARS
NKLT002	Human gar	virion prote	1.0	144-148 RRARS
NHS004	Human gar	nuclear ant	1.0	243-247 RRARS
NHS004	Human gar	virion prote	1.0	144-148 RRARS
NHS002	Human gar	nuclear ant	1.0	243-247 RRARS
NHS002	Human gar	virion prote	1.0	144-148 RRARS
HS057	Human gar	nuclear ant	1.0	243-247 RRARS
HS057	Human gar	virion prote	1.0	144-148 RRARS
HS054	Human gar	nuclear ant	1.0	243-247 RRARS
HS054	Human gar	virion prote	1.0	144-148 RRARS
HS053	Human gar	nuclear ant	1.0	243-247 RRARS
HS053	Human gar	virion prote	1.0	144-148 RRARS
HS052	Human gar	nuclear ant	1.0	243-247 RRARS
HS052	Human gar	virion prote	1.0	144-148 RRARS
HS051	Human gar	nuclear ant	1.0	243-247 RRARS
HS051	Human gar	virion prote	1.0	144-148 RRARS
HS050	Human gar	nuclear ant	1.0	243-247 RRARS
HS050	Human gar	virion prote	1.0	144-148 RRARS
HS048	Human gar	nuclear ant	1.0	243-247 RRARS
HS048	Human gar	virion prote	1.0	144-148 RRARS
HS045	Human gar	nuclear ant	1.0	243-247 RRARS
HS045	Human gar	virion prote	1.0	144-148 RRARS
HS041	Human gar	nuclear ant	1.0	243-247 RRARS
HS041	Human gar	virion prote	1.0	144-148 RRARS
HS039	Human gar	nuclear ant	1.0	243-247 RRARS
HS039	Human gar	virion prote	1.0	144-148 RRARS
HS038	Human gar	nuclear ant	1.0	243-247 RRARS
HS038	Human gar	virion prote	1.0	144-148 RRARS
HS037	Human gar	nuclear ant	1.0	243-247 RRARS

HS037	Human garvirion prote 1.0	144-148	RRARS
HS036	Human garnuclear ant 1.0	243-247	RRARS
HS036	Human garvirion prote 1.0	144-148	RRARS
HS035	Human garnuclear ant 1.0	243-247	RRARS
HS035	Human garvirion prote 1.0	144-148	RRARS
HS034	Human garnuclear ant 1.0	243-247	RRARS
HS034	Human garvirion prote 1.0	144-148	RRARS
HS033	Human garnuclear ant 1.0	243-247	RRARS
HS033	Human garvirion prote 1.0	144-148	RRARS
HS032	Human garnuclear ant 1.0	243-247	RRARS
HS032	Human garvirion prote 1.0	144-148	RRARS
HS029	Human garnuclear ant 1.0	243-247	RRARS
HS029	Human garvirion prote 1.0	144-148	RRARS
HS027	Human garnuclear ant 1.0	243-247	RRARS
HS027	Human garvirion prote 1.0	144-148	RRARS
HS025	Human garnuclear ant 1.0	243-247	RRARS
HS025	Human garvirion prote 1.0	144-148	RRARS
HS024	Human garnuclear ant 1.0	243-247	RRARS
HS024	Human garvirion prote 1.0	144-148	RRARS
HS023	Human garnuclear ant 1.0	243-247	RRARS
HS023	Human garvirion prote 1.0	144-148	RRARS
HS021	Human garnuclear ant 1.0	243-247	RRARS
HS021	Human garvirion prote 1.0	144-148	RRARS
HS020	Human garnuclear ant 1.0	243-247	RRARS
HS020	Human garvirion prote 1.0	144-148	RRARS
HS019	Human garnuclear ant 1.0	243-247	RRARS
HS019	Human garvirion prote 1.0	144-148	RRARS
HS018	Human garnuclear ant 1.0	243-247	RRARS
HS018	Human garvirion prote 1.0	144-148	RRARS
HS016	Human garnuclear ant 1.0	243-247	RRARS
HS016	Human garvirion prote 1.0	144-148	RRARS
HS015	Human garnuclear ant 1.0	243-247	RRARS
HS015	Human garvirion prote 1.0	144-148	RRARS
HS014	Human garnuclear ant 1.0	243-247	RRARS
HS014	Human garvirion prote 1.0	144-148	RRARS
HS013	Human garnuclear ant 1.0	243-247	RRARS
HS013	Human garvirion prote 1.0	144-148	RRARS
HS012	Human garnuclear ant 1.0	243-247	RRARS
HS012	Human garvirion prote 1.0	144-148	RRARS
HS011	Human garnuclear ant 1.0	243-247	RRARS
HS011	Human garvirion prote 1.0	144-148	RRARS
HS009	Human garnuclear ant 1.0	243-247	RRARS
HS009	Human garvirion prote 1.0	144-148	RRARS
HS008	Human garnuclear ant 1.0	243-247	RRARS
HS008	Human garvirion prote 1.0	144-148	RRARS
HS007	Human garnuclear ant 1.0	243-247	RRARS
HS007	Human garvirion prote 1.0	144-148	RRARS
HS003	Human garnuclear ant 1.0	243-247	RRARS
HS003	Human garvirion prote 1.0	144-148	RRARS
HS001	Human garnuclear ant 1.0	243-247	RRARS
HS001	Human garvirion prote 1.0	144-148	RRARS
HLT011	Human garnuclear ant 1.0	243-247	RRARS
HLT011	Human garvirion prote 1.0	144-148	RRARS
HLT010	Human garnuclear ant 1.0	243-247	RRARS
HLT010	Human garvirion prote 1.0	144-148	RRARS
HLT007	Human garnuclear ant 1.0	243-247	RRARS

HLT007	Human gar	virion prote	1.0	144-148	RRARS
HLT006	Human gar	nuclear ant	1.0	243-247	RRARS
HLT006	Human gar	virion prote	1.0	144-148	RRARS
HLT005	Human gar	nuclear ant	1.0	243-247	RRARS
HLT005	Human gar	virion prote	1.0	144-148	RRARS
HLT002	Human gar	nuclear ant	1.0	243-247	RRARS
HLT002	Human gar	virion prote	1.0	144-148	RRARS
HLT001	Human gar	nuclear ant	1.0	243-247	RRARS
HLT001	Human gar	virion prote	1.0	144-148	RRARS
GCT014	Human gar	nuclear ant	1.0	243-247	RRARS
GCT014	Human gar	virion prote	1.0	144-148	RRARS
GCT013	Human gar	nuclear ant	1.0	243-247	RRARS
GCT013	Human gar	virion prote	1.0	144-148	RRARS
GCT012	Human gar	nuclear ant	1.0	243-247	RRARS
GCT012	Human gar	virion prote	1.0	144-148	RRARS
GCT011	Human gar	nuclear ant	1.0	243-247	RRARS
GCT011	Human gar	virion prote	1.0	144-148	RRARS
GCT010	Human gar	nuclear ant	1.0	243-247	RRARS
GCT010	Human gar	virion prote	1.0	144-148	RRARS
GCT009	Human gar	nuclear ant	1.0	243-247	RRARS
GCT009	Human gar	virion prote	1.0	144-148	RRARS
GCT007	Human gar	nuclear ant	1.0	243-247	RRARS
GCT007	Human gar	virion prote	1.0	144-148	RRARS
GCT006	Human gar	nuclear ant	1.0	243-247	RRARS
GCT006	Human gar	virion prote	1.0	144-148	RRARS
GCT005	Human gar	nuclear ant	1.0	243-247	RRARS
GCT005	Human gar	virion prote	1.0	144-148	RRARS
GCT004	Human gar	nuclear ant	1.0	243-247	RRARS
GCT004	Human gar	virion prote	1.0	144-148	RRARS
GCT003	Human gar	nuclear ant	1.0	243-247	RRARS
GCT003	Human gar	virion prote	1.0	144-148	RRARS
GCT002	Human gar	nuclear ant	1.0	243-247	RRARS
GCT002	Human gar	virion prote	1.0	144-148	RRARS
GCT001	Human gar	nuclear ant	1.0	243-247	RRARS
GCT001	Human gar	virion prote	1.0	144-148	RRARS
C666	Human gar	nuclear ant	1.0	243-247	RRARS
C666	Human gar	virion prote	1.0	144-148	RRARS
BLT002	Human gar	nuclear ant	1.0	243-247	RRARS
BLT002	Human gar	virion prote	1.0	144-148	RRARS
BLT001	Human gar	nuclear ant	1.0	243-247	RRARS
BLT001	Human gar	virion prote	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	141-145	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	141-145	RRARS

NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	141-145	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	141-145	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
NKTCL-SG	Human gar	RPMS1 prc	1.0	61-65	RRARS
NKTCL-SG	Human gar	EBNA-3B p	1.0	243-247	RRARS
NKTCL-SG	Human gar	BLRF2 prot	1.0	144-148	RRARS
HKHD38	Human gar	RPMS1	1.0	61-65	RRARS
HKHD38	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD38	Human gar	BLRF2	1.0	144-148	RRARS
HKHD37	Human gar	RPMS1	1.0	61-65	RRARS
HKHD37	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD37	Human gar	BLRF2	1.0	144-148	RRARS
HKHD36	Human gar	RPMS1	1.0	61-65	RRARS
HKHD36	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD36	Human gar	BLRF2	1.0	144-148	RRARS
HKHD35	Human gar	RPMS1	1.0	61-65	RRARS
HKHD35	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD35	Human gar	BLRF2	1.0	144-148	RRARS
HKHD34	Human gar	RPMS1	1.0	61-65	RRARS
HKHD34	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD34	Human gar	BLRF2	1.0	144-148	RRARS
HKHD33	Human gar	RPMS1	1.0	61-65	RRARS
HKHD33	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD33	Human gar	BLRF2	1.0	141-145	RRARS
HKHD32	Human gar	RPMS1	1.0	61-65	RRARS
HKHD32	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD32	Human gar	BLRF2	1.0	141-145	RRARS
HKHD31	Human gar	RPMS1	1.0	61-65	RRARS
HKHD31	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD31	Human gar	BLRF2	1.0	141-145	RRARS
HKHD30	Human gar	RPMS1	1.0	61-65	RRARS
HKHD30	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD30	Human gar	BLRF2	1.0	141-145	RRARS
HKHD29	Human gar	EBNA-3B	1.0	231-235	RRARS
HKHD29	Human gar	BLRF2	1.0	144-148	RRARS
HKHD28	Human gar	RPMS1	1.0	61-65	RRARS
HKHD28	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD28	Human gar	BLRF2	1.0	144-148	RRARS
HKHD27	Human gar	RPMS1	1.0	61-65	RRARS
HKHD27	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD27	Human gar	BLRF2	1.0	141-145	RRARS
HKHD26	Human gar	RPMS1	1.0	61-65	RRARS

HKHD26	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD26	Human gar	BLRF2	1.0	144-148	RRARS
HKHD25	Human gar	RPMS1	1.0	61-65	RRARS
HKHD25	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD25	Human gar	BLRF2	1.0	144-148	RRARS
HKHD24	Human gar	RPMS1	1.0	61-65	RRARS
HKHD24	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD24	Human gar	BLRF2	1.0	144-148	RRARS
HKHD23	Human gar	RPMS1	1.0	61-65	RRARS
HKHD23	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD23	Human gar	BLRF2	1.0	144-148	RRARS
HKHD22	Human gar	RPMS1	1.0	61-65	RRARS
HKHD22	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD22	Human gar	BLRF2	1.0	144-148	RRARS
HKHD21	Human gar	RPMS1	1.0	61-65	RRARS
HKHD21	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD21	Human gar	BLRF2	1.0	144-148	RRARS
HKHD20	Human gar	RPMS1	1.0	61-65	RRARS
HKHD20	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD20	Human gar	BLRF2	1.0	144-148	RRARS
HKHD19	Human gar	RPMS1	1.0	61-65	RRARS
HKHD19	Human gar	BLRF2	1.0	144-148	RRARS
HKHD18	Human gar	RPMS1	1.0	61-65	RRARS
HKHD18	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD18	Human gar	BLRF2	1.0	144-148	RRARS
HKHD17	Human gar	RPMS1	1.0	61-65	RRARS
HKHD17	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD17	Human gar	BLRF2	1.0	144-148	RRARS
HKHD16	Human gar	RPMS1	1.0	61-65	RRARS
HKHD16	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD16	Human gar	BLRF2	1.0	144-148	RRARS
HKHD15	Human gar	RPMS1	1.0	61-65	RRARS
HKHD15	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD15	Human gar	BLRF2	1.0	144-148	RRARS
HKHD14	Human gar	RPMS1	1.0	61-65	RRARS
HKHD14	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD14	Human gar	BLRF2	1.0	144-148	RRARS
HKHD13	Human gar	RPMS1	1.0	61-65	RRARS
HKHD13	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD13	Human gar	BLRF2	1.0	141-145	RRARS
HKHD12	Human gar	RPMS1	1.0	61-65	RRARS
HKHD12	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD12	Human gar	BLRF2	1.0	141-145	RRARS
HKHD11	Human gar	RPMS1	1.0	61-65	RRARS
HKHD11	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD11	Human gar	BLRF2	1.0	144-148	RRARS
HKHD10	Human gar	RPMS1	1.0	61-65	RRARS
HKHD10	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD10	Human gar	BLRF2	1.0	144-148	RRARS
HKHD9	Human gar	RPMS1	1.0	61-65	RRARS
HKHD9	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD9	Human gar	BLRF2	1.0	141-145	RRARS
HKHD8	Human gar	RPMS1	1.0	61-65	RRARS
HKHD8	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD8	Human gar	BLRF2	1.0	144-148	RRARS
HKHD7	Human gar	RPMS1	1.0	61-65	RRARS

HKHD7	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD7	Human gar	BLRF2	1.0	141-145	RRARS
HKHD6	Human gar	RPMS1	1.0	61-65	RRARS
HKHD6	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD6	Human gar	BLRF2	1.0	144-148	RRARS
HKHD5	Human gar	RPMS1	1.0	61-65	RRARS
HKHD5	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD5	Human gar	BLRF2	1.0	144-148	RRARS
HKHD4	Human gar	RPMS1	1.0	61-65	RRARS
HKHD4	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD4	Human gar	BLRF2	1.0	144-148	RRARS
HKHD3	Human gar	RPMS1	1.0	61-65	RRARS
HKHD3	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD3	Human gar	BLRF2	1.0	141-145	RRARS
HKHD2	Human gar	RPMS1	1.0	61-65	RRARS
HKHD2	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD2	Human gar	BLRF2	1.0	144-148	RRARS
HKHD1	Human gar	RPMS1	1.0	61-65	RRARS
HKHD1	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD1	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC62	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC62	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC62	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC61	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC61	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC61	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC60	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC60	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC60	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC59	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC59	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC59	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC58	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC58	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC58	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC57	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC57	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC57	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC56	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC56	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC56	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC55	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC55	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC55	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC54	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC54	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC54	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC53	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC53	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC53	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC52	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC52	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC52	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC51	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC51	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC51	Human gar	BLRF2	1.0	144-148	RRARS

HKNPC50	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC50	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC50	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC49	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC49	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC48	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC48	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC48	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC47	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC47	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC47	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC46	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC46	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC46	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC45	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC45	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC45	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC44	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC44	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC44	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC43	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC43	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC43	Human gar	BLRF2	1.0	141-145	RRARS
HKNPC42	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC42	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC41	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC41	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC41	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC40	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC40	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC40	Human gar	BLRF2	1.0	141-145	RRARS
HKNPC39	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC39	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC39	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC38	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC38	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC38	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC37	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC37	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC37	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC36	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC36	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC36	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC35	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC35	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC35	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC34	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC34	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC34	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC33	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC33	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC33	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC32	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC32	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC32	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC31	Human gar	RPMS1	1.0	61-65	RRARS

HKNPC31	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC31	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC30	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC30	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC30	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC29	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC29	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC29	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC28	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC28	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC28	Human gar	BLRF2	1.0	141-145	RRARS
HKNPC27	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC27	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC27	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC26	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC26	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC26	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC25	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC25	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC25	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC24	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC24	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC24	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC23	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC23	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC23	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC22	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC22	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC22	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC21	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC21	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC21	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC20	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC20	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC20	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC19	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC19	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC19	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC18	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC18	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC18	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC17	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC17	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC17	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC16	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC16	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC16	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC15	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC15	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC15	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC14	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC14	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC14	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC13	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC13	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC13	Human gar	BLRF2	1.0	144-148	RRARS

HKNPC12	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC12	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC12	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC11	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC11	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC11	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC10	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC10	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC10	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC9	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC9	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC9	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC8	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC8	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC8	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC7	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC7	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC7	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC6	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC6	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC6	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC5	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC5	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC5	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC4	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC4	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC4	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC3	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC3	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC3	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC2	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC2	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC2	Human gar	BLRF2	1.0	144-148	RRARS
HKNPC1	Human gar	RPMS1	1.0	61-65	RRARS
HKNPC1	Human gar	EBNA-3B	1.0	243-247	RRARS
HKNPC1	Human gar	BLRF2	1.0	144-148	RRARS
HKHD142	Human gar	RPMS1	1.0	61-65	RRARS
HKHD142	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD142	Human gar	BLRF2	1.0	144-148	RRARS
HKHD141	Human gar	RPMS1	1.0	61-65	RRARS
HKHD141	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD141	Human gar	BLRF2	1.0	144-148	RRARS
HKHD140	Human gar	RPMS1	1.0	61-65	RRARS
HKHD140	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD140	Human gar	BLRF2	1.0	144-148	RRARS
HKHD139	Human gar	RPMS1	1.0	61-65	RRARS
HKHD139	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD139	Human gar	BLRF2	1.0	141-145	RRARS
HKHD138	Human gar	RPMS1	1.0	61-65	RRARS
HKHD138	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD138	Human gar	BLRF2	1.0	141-145	RRARS
HKHD137	Human gar	RPMS1	1.0	61-65	RRARS
HKHD137	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD137	Human gar	BLRF2	1.0	144-148	RRARS
HKHD136	Human gar	RPMS1	1.0	61-65	RRARS
HKHD136	Human gar	EBNA-3B	1.0	243-247	RRARS

HKHD136	Human gar	BLRF2	1.0	141-145	RRARS
HKHD135	Human gar	RPMS1	1.0	61-65	RRARS
HKHD135	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD135	Human gar	BLRF2	1.0	144-148	RRARS
HKHD134	Human gar	RPMS1	1.0	61-65	RRARS
HKHD134	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD134	Human gar	BLRF2	1.0	144-148	RRARS
HKHD133	Human gar	RPMS1	1.0	61-65	RRARS
HKHD133	Human gar	BLRF2	1.0	144-148	RRARS
HKHD132	Human gar	RPMS1	1.0	61-65	RRARS
HKHD132	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD132	Human gar	BLRF2	1.0	144-148	RRARS
HKHD131	Human gar	RPMS1	1.0	61-65	RRARS
HKHD131	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD131	Human gar	BLRF2	1.0	144-148	RRARS
HKHD130	Human gar	RPMS1	1.0	61-65	RRARS
HKHD130	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD130	Human gar	BLRF2	1.0	144-148	RRARS
HKHD129	Human gar	RPMS1	1.0	61-65	RRARS
HKHD129	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD129	Human gar	BLRF2	1.0	144-148	RRARS
HKHD128	Human gar	RPMS1	1.0	61-65	RRARS
HKHD128	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD128	Human gar	BLRF2	1.0	144-148	RRARS
HKHD127	Human gar	RPMS1	1.0	61-65	RRARS
HKHD127	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD127	Human gar	BLRF2	1.0	144-148	RRARS
HKHD126	Human gar	RPMS1	1.0	61-65	RRARS
HKHD126	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD126	Human gar	BLRF2	1.0	144-148	RRARS
HKHD125	Human gar	RPMS1	1.0	61-65	RRARS
HKHD125	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD125	Human gar	BLRF2	1.0	144-148	RRARS
HKHD124	Human gar	RPMS1	1.0	61-65	RRARS
HKHD124	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD124	Human gar	BLRF2	1.0	141-145	RRARS
HKHD123	Human gar	RPMS1	1.0	61-65	RRARS
HKHD123	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD123	Human gar	BLRF2	1.0	144-148	RRARS
HKHD122	Human gar	RPMS1	1.0	61-65	RRARS
HKHD122	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD122	Human gar	BLRF2	1.0	144-148	RRARS
HKHD121	Human gar	RPMS1	1.0	61-65	RRARS
HKHD121	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD121	Human gar	BLRF2	1.0	144-148	RRARS
HKHD120	Human gar	RPMS1	1.0	61-65	RRARS
HKHD120	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD120	Human gar	BLRF2	1.0	144-148	RRARS
HKHD119	Human gar	RPMS1	1.0	61-65	RRARS
HKHD119	Human gar	BLRF2	1.0	144-148	RRARS
HKHD118	Human gar	RPMS1	1.0	61-65	RRARS
HKHD118	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD118	Human gar	BLRF2	1.0	144-148	RRARS
HKHD117	Human gar	RPMS1	1.0	61-65	RRARS
HKHD117	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD117	Human gar	BLRF2	1.0	144-148	RRARS

HKHD116	Human gar	RPMS1	1.0	61-65	RRARS
HKHD116	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD116	Human gar	BLRF2	1.0	144-148	RRARS
HKHD115	Human gar	RPMS1	1.0	61-65	RRARS
HKHD115	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD115	Human gar	BLRF2	1.0	141-145	RRARS
HKHD114	Human gar	RPMS1	1.0	61-65	RRARS
HKHD114	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD114	Human gar	BLRF2	1.0	144-148	RRARS
HKHD113	Human gar	RPMS1	1.0	61-65	RRARS
HKHD113	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD113	Human gar	BLRF2	1.0	144-148	RRARS
HKHD112	Human gar	RPMS1	1.0	61-65	RRARS
HKHD112	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD112	Human gar	BLRF2	1.0	141-145	RRARS
HKHD111	Human gar	RPMS1	1.0	61-65	RRARS
HKHD111	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD111	Human gar	BLRF2	1.0	144-148	RRARS
HKHD110	Human gar	RPMS1	1.0	61-65	RRARS
HKHD110	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD110	Human gar	BLRF2	1.0	144-148	RRARS
HKHD109	Human gar	RPMS1	1.0	61-65	RRARS
HKHD109	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD109	Human gar	BLRF2	1.0	141-145	RRARS
HKHD108	Human gar	RPMS1	1.0	61-65	RRARS
HKHD108	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD108	Human gar	BLRF2	1.0	144-148	RRARS
HKHD107	Human gar	RPMS1	1.0	61-65	RRARS
HKHD107	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD107	Human gar	BLRF2	1.0	144-148	RRARS
HKHD106	Human gar	RPMS1	1.0	61-65	RRARS
HKHD106	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD106	Human gar	BLRF2	1.0	144-148	RRARS
HKHD105	Human gar	RPMS1	1.0	61-65	RRARS
HKHD105	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD105	Human gar	BLRF2	1.0	144-148	RRARS
HKHD104	Human gar	RPMS1	1.0	61-65	RRARS
HKHD104	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD104	Human gar	BLRF2	1.0	144-148	RRARS
HKHD103	Human gar	RPMS1	1.0	61-65	RRARS
HKHD103	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD103	Human gar	BLRF2	1.0	144-148	RRARS
HKHD102	Human gar	RPMS1	1.0	61-65	RRARS
HKHD102	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD102	Human gar	BLRF2	1.0	144-148	RRARS
HKHD101	Human gar	RPMS1	1.0	61-65	RRARS
HKHD101	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD101	Human gar	BLRF2	1.0	144-148	RRARS
HKHD100	Human gar	RPMS1	1.0	61-65	RRARS
HKHD100	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD100	Human gar	BLRF2	1.0	144-148	RRARS
HKHD99	Human gar	RPMS1	1.0	61-65	RRARS
HKHD99	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD99	Human gar	BLRF2	1.0	141-145	RRARS
HKHD98	Human gar	RPMS1	1.0	61-65	RRARS
HKHD98	Human gar	EBNA-3B	1.0	243-247	RRARS

HKHD98	Human gar	BLRF2	1.0	144-148	RRARS
HKHD97	Human gar	RPMS1	1.0	61-65	RRARS
HKHD97	Human gar	BLRF2	1.0	144-148	RRARS
HKHD96	Human gar	RPMS1	1.0	61-65	RRARS
HKHD96	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD96	Human gar	BLRF2	1.0	144-148	RRARS
HKHD95	Human gar	RPMS1	1.0	61-65	RRARS
HKHD95	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD95	Human gar	BLRF2	1.0	144-148	RRARS
HKHD94	Human gar	RPMS1	1.0	61-65	RRARS
HKHD94	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD94	Human gar	BLRF2	1.0	144-148	RRARS
HKHD93	Human gar	RPMS1	1.0	61-65	RRARS
HKHD93	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD93	Human gar	BLRF2	1.0	144-148	RRARS
HKHD92	Human gar	RPMS1	1.0	61-65	RRARS
HKHD92	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD92	Human gar	BLRF2	1.0	144-148	RRARS
HKHD91	Human gar	RPMS1	1.0	61-65	RRARS
HKHD91	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD91	Human gar	BLRF2	1.0	144-148	RRARS
HKHD90	Human gar	RPMS1	1.0	61-65	RRARS
HKHD90	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD90	Human gar	BLRF2	1.0	144-148	RRARS
HKHD89	Human gar	RPMS1	1.0	61-65	RRARS
HKHD89	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD89	Human gar	BLRF2	1.0	141-145	RRARS
HKHD88	Human gar	RPMS1	1.0	61-65	RRARS
HKHD88	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD88	Human gar	BLRF2	1.0	144-148	RRARS
HKHD87	Human gar	RPMS1	1.0	61-65	RRARS
HKHD87	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD87	Human gar	BLRF2	1.0	144-148	RRARS
HKHD86	Human gar	RPMS1	1.0	61-65	RRARS
HKHD86	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD86	Human gar	BLRF2	1.0	144-148	RRARS
HKHD85	Human gar	RPMS1	1.0	61-65	RRARS
HKHD85	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD85	Human gar	BLRF2	1.0	144-148	RRARS
HKHD84	Human gar	RPMS1	1.0	61-65	RRARS
HKHD84	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD84	Human gar	BLRF2	1.0	141-145	RRARS
HKHD83	Human gar	RPMS1	1.0	61-65	RRARS
HKHD83	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD83	Human gar	BLRF2	1.0	141-145	RRARS
HKHD82	Human gar	RPMS1	1.0	61-65	RRARS
HKHD82	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD82	Human gar	BLRF2	1.0	144-148	RRARS
HKHD81	Human gar	RPMS1	1.0	61-65	RRARS
HKHD81	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD81	Human gar	BLRF2	1.0	144-148	RRARS
HKHD80	Human gar	RPMS1	1.0	61-65	RRARS
HKHD80	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD80	Human gar	BLRF2	1.0	141-145	RRARS
HKHD79	Human gar	RPMS1	1.0	61-65	RRARS
HKHD79	Human gar	EBNA-3B	1.0	243-247	RRARS

HKHD79	Human gar	BLRF2	1.0	144-148	RRARS
HKHD78	Human gar	BLRF2	1.0	144-148	RRARS
HKHD78	Human gar	RPMS1	1.0	61-65	RRARS
HKHD78	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD77	Human gar	RPMS1	1.0	61-65	RRARS
HKHD77	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD77	Human gar	BLRF2	1.0	144-148	RRARS
HKHD76	Human gar	RPMS1	1.0	61-65	RRARS
HKHD76	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD76	Human gar	BLRF2	1.0	141-145	RRARS
HKHD75	Human gar	RPMS1	1.0	61-65	RRARS
HKHD75	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD75	Human gar	BLRF2	1.0	144-148	RRARS
HKHD74	Human gar	RPMS1	1.0	61-65	RRARS
HKHD74	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD74	Human gar	BLRF2	1.0	141-145	RRARS
HKHD73	Human gar	RPMS1	1.0	61-65	RRARS
HKHD73	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD73	Human gar	BLRF2	1.0	144-148	RRARS
HKHD72	Human gar	RPMS1	1.0	61-65	RRARS
HKHD72	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD72	Human gar	BLRF2	1.0	144-148	RRARS
HKHD71	Human gar	RPMS1	1.0	61-65	RRARS
HKHD71	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD71	Human gar	BLRF2	1.0	144-148	RRARS
HKHD70	Human gar	RPMS1	1.0	61-65	RRARS
HKHD70	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD70	Human gar	BLRF2	1.0	141-145	RRARS
HKHD69	Human gar	RPMS1	1.0	61-65	RRARS
HKHD69	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD69	Human gar	BLRF2	1.0	144-148	RRARS
HKHD68	Human gar	RPMS1	1.0	61-65	RRARS
HKHD68	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD68	Human gar	BLRF2	1.0	144-148	RRARS
HKHD67	Human gar	RPMS1	1.0	61-65	RRARS
HKHD67	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD67	Human gar	BLRF2	1.0	144-148	RRARS
HKHD66	Human gar	RPMS1	1.0	61-65	RRARS
HKHD66	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD66	Human gar	BLRF2	1.0	141-145	RRARS
HKHD65	Human gar	RPMS1	1.0	61-65	RRARS
HKHD65	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD65	Human gar	BLRF2	1.0	144-148	RRARS
HKHD64	Human gar	RPMS1	1.0	61-65	RRARS
HKHD64	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD64	Human gar	BLRF2	1.0	144-148	RRARS
HKHD63	Human gar	RPMS1	1.0	61-65	RRARS
HKHD63	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD63	Human gar	BLRF2	1.0	144-148	RRARS
HKHD62	Human gar	RPMS1	1.0	61-65	RRARS
HKHD62	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD62	Human gar	BLRF2	1.0	144-148	RRARS
HKHD61	Human gar	RPMS1	1.0	61-65	RRARS
HKHD61	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD61	Human gar	BLRF2	1.0	141-145	RRARS
HKHD60	Human gar	RPMS1	1.0	61-65	RRARS

HKHD60	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD60	Human gar	BLRF2	1.0	141-145	RRARS
HKHD59	Human gar	RPMS1	1.0	61-65	RRARS
HKHD59	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD59	Human gar	BLRF2	1.0	141-145	RRARS
HKHD58	Human gar	RPMS1	1.0	61-65	RRARS
HKHD58	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD58	Human gar	BLRF2	1.0	144-148	RRARS
HKHD57	Human gar	RPMS1	1.0	61-65	RRARS
HKHD57	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD57	Human gar	BLRF2	1.0	144-148	RRARS
HKHD56	Human gar	RPMS1	1.0	61-65	RRARS
HKHD56	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD56	Human gar	BLRF2	1.0	141-145	RRARS
HKHD55	Human gar	RPMS1	1.0	61-65	RRARS
HKHD55	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD55	Human gar	BLRF2	1.0	141-145	RRARS
HKHD54	Human gar	RPMS1	1.0	61-65	RRARS
HKHD54	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD54	Human gar	BLRF2	1.0	144-148	RRARS
HKHD53	Human gar	RPMS1	1.0	61-65	RRARS
HKHD53	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD53	Human gar	BLRF2	1.0	144-148	RRARS
HKHD52	Human gar	RPMS1	1.0	61-65	RRARS
HKHD52	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD52	Human gar	BLRF2	1.0	144-148	RRARS
HKHD51	Human gar	RPMS1	1.0	61-65	RRARS
HKHD51	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD51	Human gar	BLRF2	1.0	141-145	RRARS
HKHD50	Human gar	RPMS1	1.0	61-65	RRARS
HKHD50	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD50	Human gar	BLRF2	1.0	141-145	RRARS
HKHD49	Human gar	RPMS1	1.0	61-65	RRARS
HKHD49	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD49	Human gar	BLRF2	1.0	141-145	RRARS
HKHD48	Human gar	RPMS1	1.0	61-65	RRARS
HKHD48	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD48	Human gar	BLRF2	1.0	144-148	RRARS
HKHD47	Human gar	RPMS1	1.0	61-65	RRARS
HKHD47	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD47	Human gar	BLRF2	1.0	144-148	RRARS
HKHD46	Human gar	RPMS1	1.0	61-65	RRARS
HKHD46	Human gar	BLRF2	1.0	141-145	RRARS
HKHD45	Human gar	RPMS1	1.0	61-65	RRARS
HKHD45	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD45	Human gar	BLRF2	1.0	144-148	RRARS
HKHD44	Human gar	RPMS1	1.0	61-65	RRARS
HKHD44	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD44	Human gar	BLRF2	1.0	141-145	RRARS
HKHD43	Human gar	RPMS1	1.0	61-65	RRARS
HKHD43	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD43	Human gar	BLRF2	1.0	141-145	RRARS
HKHD42	Human gar	RPMS1	1.0	61-65	RRARS
HKHD42	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD42	Human gar	BLRF2	1.0	144-148	RRARS
HKHD41	Human gar	RPMS1	1.0	61-65	RRARS

HKHD41	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD41	Human gar	BLRF2	1.0	144-148	RRARS
HKHD40	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD40	Human gar	BLRF2	1.0	144-148	RRARS
HKHD39	Human gar	RPMS1	1.0	61-65	RRARS
HKHD39	Human gar	EBNA-3B	1.0	243-247	RRARS
HKHD39	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-2.22	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-2.22	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-2.22	Human gar	BLRF2	1.0	144-148	RRARS
Jijoye	Human gar	RPMS1	1.0	61-65	RRARS
Jijoye	Human gar	EBNA-3B	1.0	243-247	RRARS
Jijoye	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IM1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IM1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IM1.1	Human gar	BLRF2	1.0	144-148	RRARS
LCL_B958_	Human gar	EBNA-3B	1.0	243-247	RRARS
LCL_B958_	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IM1.(	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IM1.(	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IM1.(	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-IS1.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-2.15	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-2.15	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-2.15	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IM1.(	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IM1.(	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IM1.(	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-IS2.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS2.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS2.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-2.21	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-2.21	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-2.21	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-1.04	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.04	Human gar	EBNA-3B	1.0	243-247	RRARS

sLCL-1.04	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IM1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IM1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IM1.1	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-BL1.(	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-BL1.(	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-BL1.(	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.05	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.05	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.05	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-2.16	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-2.16	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-2.16	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.13	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.13	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.13	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-1.17	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.17	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.17	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-IS1.2	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.2	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.2	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-1.09	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.09	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.09	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.10	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.10	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.10	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-BL1.2	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-BL1.2	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-BL1.2	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.1	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.1	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.1	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.24	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.24	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.24	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IM1.(	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IM1.(	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IM1.(	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-1.06	Human gar	RPMS1	1.0	61-65	RRARS

sLCL-1.06	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.06	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.07	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.07	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.07	Human gar	BLRF2	1.0	144-148	RRARS
HL04	Human gar	RPMS1	1.0	61-65	RRARS
HL04	Human gar	EBNA-3B	1.0	243-247	RRARS
HL04	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-1.18	Human her	RPMS1	1.0	61-65	RRARS
sLCL-1.18	Human her	EBNA-3B	1.0	243-247	RRARS
sLCL-1.18	Human her	BLRF2	1.0	144-148	RRARS
sLCL-1.19	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.19	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.19	Human gar	BLRF2	1.0	141-145	RRARS
YCCEL1	Human gar	RPMS1	1.0	61-65	RRARS
YCCEL1	Human gar	EBNA-3B	1.0	243-247	RRARS
YCCEL1	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-2.14	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-2.14	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-2.14	Human gar	BLRF2	1.0	144-148	RRARS
pLCL-TRL5	Human gar	RPMS1	1.0	61-65	RRARS
pLCL-TRL5	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.02	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.02	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.02	Human gar	BLRF2	1.0	144-148	RRARS
BL36	Human her	RPMS1	1.0	61-65	RRARS
BL36	Human her	EBNA-3B	1.0	243-247	RRARS
BL36	Human her	BLRF2	1.0	144-148	RRARS
Cheptages	Human gar	RPMS1	1.0	61-65	RRARS
Cheptages	Human gar	EBNA-3B	1.0	243-247	RRARS
Cheptages	Human gar	BLRF2	1.0	144-148	RRARS
X50-7	Human gar	RPMS1	1.0	61-65	RRARS
X50-7	Human gar	BLRF2	1.0	144-148	RRARS
AFB1	Human gar	RPMS1	1.0	61-65	RRARS
AFB1	Human gar	EBNA-3B	1.0	243-247	RRARS
AFB1	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-IS1.0	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-IS1.0	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-IS1.0	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.08	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.08	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.08	Human gar	BLRF2	1.0	144-148	RRARS
Makau	Human gar	RPMS1	1.0	61-65	RRARS
Makau	Human gar	EBNA-3B	1.0	243-247	RRARS
Makau	Human gar	BLRF2	1.0	141-145	RRARS
sLCL-1.11	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.11	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.11	Human gar	BLRF2	1.0	144-148	RRARS
D3201.2	Human gar	RPMS1	1.0	61-65	RRARS
D3201.2	Human gar	EBNA-3B	1.0	243-247	RRARS
D3201.2	Human gar	BLRF2	1.0	144-148	RRARS
P3HR1_c1(	Human her	RPMS1	1.0	61-65	RRARS
P3HR1_c1(	Human her	EBNA-3B	1.0	243-247	RRARS
P3HR1_c1(	Human her	BLRF2	1.0	144-148	RRARS
HKN15	Human gar	RPMS1	1.0	61-65	RRARS
HKN15	Human gar	EBNA-3B	1.0	243-247	RRARS

HKN15	Human gar	BLRF2	1.0	144-148	RRARS
HL02	Human gar	RPMS1	1.0	61-65	RRARS
HL02	Human gar	EBNA-3B	1.0	243-247	RRARS
HL02	Human gar	BLRF2	1.0	144-148	RRARS
Daudi	Human gar	RPMS1	1.0	61-65	RRARS
Daudi	Human gar	EBNA-3B	1.0	243-247	RRARS
Daudi	Human gar	BLRF2	1.0	144-148	RRARS
Wewak_2	Human gar	RPMS1	1.0	61-65	RRARS
Wewak_2	Human gar	EBNA-3B	1.0	243-247	RRARS
Wewak_2	Human gar	BLRF2	1.0	141-145	RRARS
M-ABA	Human gar	RPMS1	1.0	61-65	RRARS
M-ABA	Human gar	EBNA-3B	1.0	243-247	RRARS
M-ABA	Human gar	BLRF2	1.0	141-145	RRARS
BL37	Human gar	RPMS1	1.0	61-65	RRARS
BL37	Human gar	EBNA-3B	1.0	243-247	RRARS
BL37	Human gar	BLRF2	1.0	141-145	RRARS
C666-1	Human gar	RPMS1	1.0	61-65	RRARS
C666-1	Human gar	EBNA-3B	1.0	243-247	RRARS
C666-1	Human gar	BLRF2	1.0	144-148	RRARS
HL11	Human gar	RPMS1	1.0	61-65	RRARS
HL11	Human gar	EBNA-3B	1.0	243-247	RRARS
HL11	Human gar	BLRF2	1.0	144-148	RRARS
L591	Human gar	RPMS1	1.0	61-65	RRARS
L591	Human gar	EBNA-3B	1.0	243-247	RRARS
L591	Human gar	BLRF2	1.0	144-148	RRARS
HL09	Human gar	RPMS1	1.0	61-65	RRARS
HL09	Human gar	EBNA-3B	1.0	243-247	RRARS
HL09	Human gar	BLRF2	1.0	141-145	RRARS
HL01	Human gar	RPMS1	1.0	61-65	RRARS
HL01	Human gar	EBNA-3B	1.0	243-247	RRARS
HL01	Human gar	BLRF2	1.0	144-148	RRARS
HL08	Human gar	RPMS1	1.0	61-65	RRARS
HL08	Human gar	EBNA-3B	1.0	243-247	RRARS
HL08	Human gar	BLRF2	1.0	144-148	RRARS
HKN19	Human gar	RPMS1	1.0	61-65	RRARS
HKN19	Human gar	EBNA-3B	1.0	243-247	RRARS
HKN19	Human gar	BLRF2	1.0	144-148	RRARS
HKN14	Human gar	RPMS1	1.0	61-65	RRARS
HKN14	Human gar	EBNA-3B	1.0	243-247	RRARS
HKN14	Human gar	BLRF2	1.0	141-145	RRARS
Akata	Human gar	RPMS1	1.0	61-65	RRARS
Akata	Human gar	EBNA-3B	1.0	243-247	RRARS
Akata	Human gar	BLRF2	1.0	144-148	RRARS
pLCL-TRL1	Human gar	RPMS1	1.0	61-65	RRARS
pLCL-TRL1	Human gar	BLRF2	1.0	144-148	RRARS
pLCL-TRL1	Human gar	RPMS1	1.0	61-65	RRARS
pLCL-TRL1	Human gar	BLRF2	1.0	144-148	RRARS
sLCL-1.12	Human gar	RPMS1	1.0	61-65	RRARS
sLCL-1.12	Human gar	EBNA-3B	1.0	243-247	RRARS
sLCL-1.12	Human gar	BLRF2	1.0	141-145	RRARS
HL05	Human gar	RPMS1	1.0	61-65	RRARS
HL05	Human gar	EBNA-3B	1.0	243-247	RRARS
HL05	Human gar	BLRF2	1.0	141-145	RRARS
EBV Mak_´	Human gar	RPMS1	1.0	61-65	RRARS
EBV Mak_´	Human gar	EBNA-3B	1.0	243-247	RRARS
EBV Mak_´	Human gar	BLRF2	1.0	141-145	RRARS

UNKNOWN	Human gar	RPMS1	1.0	61-65	RRARS
UNKNOWN	Human gar	EBNA-3B	1.0	243-247	RRARS
UNKNOWN	Human gar	BLRF2	1.0	144-148	RRARS
B95-8	Human her	RPMS1 prc	1.0	61-65	RRARS
B95-8	Human her	EBNA-3B n	1.0	243-247	RRARS
B95-8	Human her	putative BL	1.0	144-148	RRARS
C15	Human her	RPMS1 prc	1.0	61-65	RRARS
B95-8	Human her	EBNA3B (E	1.0	243-247	RRARS
B95-8	Human her	-N/A-	1.0	144-148	RRARS
B 95-8	Human her	COMPLETI	1.0	144-148	RRARS
HNNPC8	Human gar	putative BL	1.0	128-132	RRARS
HNNPC7	Human gar	putative BL	1.0	128-132	RRARS
HNNPC6	Human gar	putative BL	1.0	128-132	RRARS
HNNPC5	Human gar	putative BL	1.0	128-132	RRARS
HNNPC4	Human gar	putative BL	1.0	128-132	RRARS
HNNPC3	Human gar	putative BL	1.0	128-132	RRARS
HNNPC2	Human gar	putative BL	1.0	128-132	RRARS
HNNPC1	Human her	putative BL	1.0	128-132	RRARS
YCCEL1	Human her	RPMS1 prc	1.0	61-65	RRARS
YCCEL1	Human her	EBNA-3B n	1.0	243-247	RRARS
YCCEL1	Human her	putative BL	1.0	144-148	RRARS
SNU-719	Human her	RPMS1 prc	1.0	61-65	RRARS
SNU-719	Human her	EBNA-3B n	1.0	243-247	RRARS
SNU-719	Human her	putative BL	1.0	141-145	RRARS
HN18	Human her	tegument p	1.0	144-148	RRARS
HN9	Human her	tegument p	1.0	144-148	RRARS
HN8	Human her	tegument p	1.0	144-148	RRARS
HN15	Human her	-N/A-	1.0	61-65	RRARS
HN15	Human her	EBNA3B nu	1.0	243-247	RRARS
HN15	Human her	tegument p	1.0	144-148	RRARS
HN6	Human her	tegument p	1.0	144-148	RRARS
HN5	Human her	tegument p	1.0	144-148	RRARS
HN14	Human her	tegument p	1.0	144-148	RRARS
HN13	Human her	tegument p	1.0	144-148	RRARS
HN4	Human her	tegument p	1.0	144-148	RRARS
HN3	Human her	tegument p	1.0	144-148	RRARS
HN12	Human her	-N/A-	1.0	61-65	RRARS
HN12	Human her	tegument p	1.0	144-148	RRARS
HN11	Human her	tegument p	1.0	144-148	RRARS
HN10	Human her	tegument p	1.0	144-148	RRARS
HN2	Human her	tegument p	1.0	141-145	RRARS
HN1	Human her	-N/A-	1.0	61-65	RRARS
HN1	Human her	tegument p	1.0	144-148	RRARS
1 LGY-Raji	Human her	EBNA3B nu	1.0	243-247	RRARS
1 LGY-Raji	Human her	tegument p	1.0	144-148	RRARS
1 LGY-C66	Human her	tegument p	1.0	144-148	RRARS
HVMA	Macaca ar	hypothetica	1.0	145-149	RRARS
P3-T1	Human gar	BLRF2	1.0	144-148	RRARS
P2-T1	Human gar	BLRF2	1.0	144-148	RRARS
ebv27	Human gar	BLRF2	1.0	144-148	RRARS
ebv25	Human gar	BLRF2	1.0	144-148	RRARS
ebv22	Human gar	BLRF2	1.0	144-148	RRARS
ebv17	Human gar	BLRF2	1.0	144-148	RRARS
ebv9	Human gar	BLRF2	1.0	144-148	RRARS
ebv8	Human gar	BLRF2	1.0	144-148	RRARS
ebv7	Human gar	BLRF2	1.0	144-148	RRARS

Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
Epstein-Ba	Human gar	BLRF2	1.0	144-148	RRARS
M81_gDNA	Human gar	BLRF2	1.0	144-148	RRARS
JWBL43B	Human gar	BLRF2	1.0	144-148	RRARS
JWBL121B	Human gar	BLRF2	1.0	144-148	RRARS
JM_Saliva_	Human gar	BLRF2	1.0	144-148	RRARS
JM_Saliva_	Human gar	BLRF2	1.0	144-148	RRARS
JM_Saliva_	Human gar	BLRF2	1.0	144-148	RRARS
JM_NPC_b	Human gar	BLRF2	1.0	144-148	RRARS
JM_NPC_b	Human gar	BLRF2	1.0	144-148	RRARS
JM_NPC_b	Human gar	BLRF2	1.0	144-148	RRARS
JM_LCL_Ik	Human gar	BLRF2	1.0	144-148	RRARS
JC_037	Human gar	BLRF2	1.0	144-148	RRARS
JC_030_29	Human gar	BLRF2	1.0	144-148	RRARS
JC_030_18	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
IMS_Saliva	Human gar	BLRF2	1.0	144-148	RRARS
GK_RUDU	Human gar	BLRF2	1.0	144-148	RRARS
GK_LY65	Human gar	BLRF2	1.0	144-148	RRARS
GK_LY47	Human gar	BLRF2	1.0	144-148	RRARS
GK_BL67	Human gar	BLRF2	1.0	144-148	RRARS
GK_BL60	Human gar	BLRF2	1.0	144-148	RRARS
GK_BL44	Human gar	BLRF2	1.0	144-148	RRARS
GK_BL42	Human gar	BLRF2	1.0	144-148	RRARS
GK_BL36	Human gar	BLRF2	1.0	144-148	RRARS
GK_BL16	Human gar	BLRF2	1.0	144-148	RRARS
GK_Akuba	Human gar	BLRF2	1.0	144-148	RRARS
DF_Tonsil_	Human gar	BLRF2	1.0	144-148	RRARS
DF_Tonsil_	Human gar	BLRF2	1.0	144-148	RRARS
AH_Saliva_	Human gar	BLRF2	1.0	144-148	RRARS
EBVaGC5-	Human gar	RPMS1 prc	1.0	61-65	RRARS
EBVaGC5-	Human gar	EBNA3B nu	1.0	243-247	RRARS
EBVaGC5-	Human gar	BLRF2 prot	1.0	144-148	RRARS
HKNPC6-G	Human gar	EBNA3B nu	1.0	243-247	RRARS
HKNPC6-G	Human gar	BLRF2 prot	1.0	141-145	RRARS
EBVaGC8-	Human gar	RPMS1 prc	1.0	61-65	RRARS
EBVaGC8-	Human gar	EBNA3B nu	1.0	243-247	RRARS
EBVaGC8-	Human gar	BLRF2 prot	1.0	141-145	RRARS
EBVaGC8-	Human gar	RPMS1 prc	1.0	61-65	RRARS
EBVaGC8-	Human gar	EBNA3B nu	1.0	243-247	RRARS
EBVaGC8-	Human gar	BLRF2 prot	1.0	141-145	RRARS
EBVaGC8-	Human gar	RPMS1 prc	1.0	61-65	RRARS
EBVaGC8-	Human gar	EBNA3B nu	1.0	243-247	RRARS
EBVaGC8-	Human gar	BLRF2 prot	1.0	144-148	RRARS

AG876-GC	Human gar	EBNA3B nu	1.0	243-247	RRARS
AG876-GC	Human gar	BLRF2 prot	1.0	144-148	RRARS
Mutu-GC4	Human gar	RPMS1 prc	1.0	61-65	RRARS
Mutu-GC4	Human gar	EBNA3B nu	1.0	243-247	RRARS
Mutu-GC4	Human gar	BLRF2 prot	1.0	144-148	RRARS
Mutu-GC3	Human gar	RPMS1 prc	1.0	61-65	RRARS
Mutu-GC3	Human gar	EBNA3B nu	1.0	243-247	RRARS
Mutu-GC3	Human gar	BLRF2 prot	1.0	144-148	RRARS
Mutu-GC2	Human gar	RPMS1 prc	1.0	61-65	RRARS
Mutu-GC2	Human gar	EBNA3B nu	1.0	243-247	RRARS
Mutu-GC2	Human gar	BLRF2 prot	1.0	144-148	RRARS
Mutu-GC1	Human gar	RPMS1 prc	1.0	61-65	RRARS
Mutu-GC1	Human gar	EBNA3B nu	1.0	243-247	RRARS
Mutu-GC1	Human gar	BLRF2 prot	1.0	144-148	RRARS
Akata-GC1	Human gar	RPMS1 prc	1.0	61-65	RRARS
Akata-GC1	Human gar	EBNA3B nu	1.0	243-247	RRARS
Akata-GC1	Human gar	BLRF2 prot	1.0	144-148	RRARS
YCCEL1-G	Human gar	RPMS1 prc	1.0	61-65	RRARS
YCCEL1-G	Human gar	EBNA3B nu	1.0	243-247	RRARS
YCCEL1-G	Human gar	putative BL	1.0	144-148	RRARS
YCCEL1-G	Human gar	RPMS1 prc	1.0	61-65	RRARS
YCCEL1-G	Human gar	EBNA3B nu	1.0	243-247	RRARS
YCCEL1-G	Human gar	putative BL	1.0	144-148	RRARS
YCCEL1	Human gar	RPMS1 prc	1.0	61-65	RRARS
YCCEL1	Human gar	EBNA-3B n	1.0	243-247	RRARS
YCCEL1	Human gar	putative BL	1.0	144-148	RRARS
SNU-719	Human gar	RPMS1 prc	1.0	61-65	RRARS
SNU-719	Human gar	putative BL	1.0	141-145	RRARS
GDGC2	Human gar	RPMS1 prc	1.0	61-65	RRARS
GDGC2	Human gar	EBNA-3B n	1.0	243-247	RRARS
GDGC2	Human gar	putative BL	1.0	144-148	RRARS
GDGC1	Human gar	RPMS1 prc	1.0	61-65	RRARS
GDGC1	Human gar	putative BL	1.0	144-148	RRARS
YCCEL1	Human gar	EBNA-3B n	1.0	243-247	RRARS
YCCEL1	Human gar	putative BL	1.0	144-148	RRARS
SNU-719	Human gar	EBNA-3B n	1.0	243-247	RRARS
SNU-719	Human gar	putative BL	1.0	141-145	RRARS
GC-EBV2	Human gar	EBNA-3B n	1.0	243-247	RRARS
GC-EBV2	Human gar	putative BL	1.0	144-148	RRARS
GC-EBV1	Human gar	EBNA-3B n	1.0	243-247	RRARS
GC-EBV1	Human gar	putative BL	1.0	144-148	RRARS
LC4	Human her	RPMS1 prc	1.0	61-65	RRARS
LC4	Human her	EBNA-3B n	1.0	243-247	RRARS
LC4	Human her	putative BL	1.0	144-148	RRARS
LC3	Human her	RPMS1 prc	1.0	61-65	RRARS
LC3	Human her	putative BL	1.0	144-148	RRARS
LC2	Human her	RPMS1 prc	1.0	61-65	RRARS
LC2	Human her	EBNA-3B n	1.0	243-247	RRARS
LC2	Human her	putative BL	1.0	144-148	RRARS
LC1	Human her	RPMS1 prc	1.0	61-65	RRARS
LC1	Human her	EBNA-3B n	1.0	243-247	RRARS
LC1	Human her	putative BL	1.0	144-148	RRARS
EBVaGC9	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC9	Human her	EBNA3B	1.0	243-247	RRARS
EBVaGC9	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC8	Human her	RPMS1	1.0	61-65	RRARS

EBVaGC8	Human her	EBNA3B	1.0	243-247	RRARS
EBVaGC8	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC7	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC7	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC6	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC6	Human her	EBNA3B	1.0	243-247	RRARS
EBVaGC6	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC5	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC5	Human her	EBNA3B	1.0	243-247	RRARS
EBVaGC5	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC4	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC4	Human her	EBNA-3B	1.0	243-247	RRARS
EBVaGC4	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC2	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC2	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC1	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC1	Human her	EBNA3B	1.0	243-247	RRARS
EBVaGC1	Human her	BLRF2	1.0	144-148	RRARS
EBVaGC3	Human her	RPMS1	1.0	61-65	RRARS
EBVaGC3	Human her	BLRF2	1.0	144-148	RRARS
SG	Human her	EBNA-3B n	1.0	243-247	RRARS
SG	Human her	putative BL	1.0	141-145	RRARS
VA	Human her	EBNA-3B n	1.0	243-247	RRARS
VA	Human her	putative BL	1.0	144-148	RRARS
FNR	Human her	RPMS1 prc	1.0	61-65	RRARS
FNR	Human her	EBNA-3B n	1.0	243-247	RRARS
FNR	Human her	putative BL	1.0	144-148	RRARS
RPF	Human her	RPMS1 prc	1.0	61-65	RRARS
RPF	Human her	EBNA-3B n	1.0	243-247	RRARS
RPF	Human her	putative BL	1.0	144-148	RRARS
CV-ARG	Human her	RPMS1 prc	1.0	61-65	RRARS
CV-ARG	Human her	EBNA-3B n	1.0	243-247	RRARS
CV-ARG	Human her	putative BL	1.0	144-148	RRARS
H03753A	Human her	RPMS1 prc	1.0	61-65	RRARS
H03753A	Human her	EBNA-3B n	1.0	243-247	RRARS
H03753A	Human her	putative BL	1.0	144-148	RRARS
pfe-lcl-E3	Unclassifie	BLRF2	1.0	145-149	RRARS
GC1	Human her	RPMS1 prc	1.0	61-65	RRARS
GC1	Human her	EBNA-3B n	1.0	243-247	RRARS
GC1	Human her	putative BL	1.0	144-148	RRARS
H002213	Human her	RPMS1 prc	1.0	61-65	RRARS
H002213	Human her	EBNA-3B n	1.0	243-247	RRARS
H002213	Human her	putative BL	1.0	144-148	RRARS
H058015C	Human her	RPMS1 prc	1.0	61-65	RRARS
H058015C	Human her	EBNA-3B n	1.0	243-247	RRARS
H058015C	Human her	putative BL	1.0	144-148	RRARS
H018436D	Human her	RPMS1 prc	1.0	61-65	RRARS
H018436D	Human her	EBNA-3B n	1.0	243-247	RRARS
H018436D	Human her	putative BL	1.0	144-148	RRARS
HU11393	Human her	RPMS1 prc	1.0	61-65	RRARS
HU11393	Human her	EBNA-3B n	1.0	243-247	RRARS
HU11393	Human her	putative BL	1.0	144-148	RRARS
VGO	Human her	RPMS1 prc	1.0	61-65	RRARS
VGO	Human her	EBNA-3B n	1.0	243-247	RRARS
VGO	Human her	putative BL	1.0	141-145	RRARS
SCL	Human her	RPMS1 prc	1.0	61-65	RRARS

SCL	Human her	EBNA-3B n	1.0	243-247	RRARS
SCL	Human her	putative BL	1.0	141-145	RRARS
MP	Human her	EBNA-3B n	1.0	243-247	RRARS
MP	Human her	putative BL	1.0	141-145	RRARS
CCH	Human her	EBNA-3B n	1.0	243-247	RRARS
CCH	Human her	putative BL	1.0	144-148	RRARS
Raji	Human her	EBNA-3b	1.0	243-247	RRARS
Raji	Human her	BLRF2	1.0	144-148	RRARS
C666-1	Human her	EBNA-3B	1.0	243-247	RRARS
C666-1	Human her	BLRF2	1.0	144-148	RRARS
HKNPC9	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC9	Human her	BLRF2	1.0	144-148	RRARS
HKNPC8	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC8	Human her	BLRF2	1.0	144-148	RRARS
HKNPC7	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC7	Human her	BLRF2	1.0	144-148	RRARS
HKNPC6	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC6	Human her	BLRF2	1.0	144-148	RRARS
HKNPC5	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC5	Human her	BLRF2	1.0	144-148	RRARS
HKNPC4	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC4	Human her	BLRF2	1.0	144-148	RRARS
HKNPC3	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC3	Human her	BLRF2	1.0	144-148	RRARS
HKNPC2	Human her	EBNA-3B	1.0	243-247	RRARS
HKNPC2	Human her	BLRF2	1.0	144-148	RRARS
K4413-Mi	Human her	latency pro	1.0	243-247	RRARS
C666-1	Human her	EBNA-3B/E	1.0	243-247	RRARS
C666-1	Human her	BLRF2	1.0	144-148	RRARS
Akata	Human her	RPMS1	1.0	61-65	RRARS
Akata	Human her	BLRF2	1.0	144-148	RRARS
Akata	Human her	EBNA-3B	1.0	243-247	RRARS
GD2	Human her	EBNA3B	1.0	243-247	RRARS
GD2	Human her	BLRF2	1.0	144-148	RRARS
AG876	Human her	EBNA-3B	1.0	243-247	RRARS
AG876	Human her	BLRF2	1.0	144-148	RRARS
GD1	Human her	unknown	1.0	61-65	RRARS
GD1	Human her	EBNA3B (E	1.0	243-247	RRARS
GD1	Human her	unknown	1.0	144-148	RRARS
MISP	Human her	nuclear ant	1.0	243-247	RRARS
LCL8664	Macacine h	BLRF2	1.0	145-149	RRARS
UNKNOWN	Human her	-N/A-	1.0	243-247	RRARS
2715	Cercopithe	protein UL1	1.0	168-172	RRARS
Colburn	Cercopithe	protein UL1	1.0	168-172	RRARS
2715	Cercopithe	protein UL1	1.0	168-172	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
RP-12	Myodes narm	nucleocapsid	1.0	443-447	RRARS
GUINEA BISS	Peste-des-pet	nucleocapsid	1.0	383-387	RRARS
88	Peste-des-pet	nucleoprotein	1.0	34-38	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
EgB 90	Tete orthobun	polyprotein	1.0	726-730	RRARS
EgB 90	Tete orthobun	polyprotein	1.0	726-730	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
Izmir 19	Unclassified P	polymerase	1.0	972-976	RRARS
Ethiopia-2011	Unclassified P	RNA-depende	1.0	972-976	RRARS
Izmir 19	Unclassified P	polymerase	1.0	972-976	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
MW07	Unclassified M	polyprotein	1.0	989-993	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
Red squirrel U	Squirrelpox vir	intermediate t	1.0	282-286	RRARS
Red squirrel U	Squirrelpox vir	intermediate t	1.0	282-286	RRARS
HL953	Parapoxvirus	hypothetical p	1.0	398-402	RRARS
GO	Orf virus	hypothetical p	1.0	18-22	RRARS
HL953	Unclassified P	hypothetical p	1.0	398-402	RRARS
D1701	Orf virus	PP184	1.0	52-56	RRARS
FeP2	Pigeonpox vir	DNA-binding v	1.0	122-126	RRARS
PSan92	Unclassified A	DNA-binding v	1.0	122-126	RRARS
PSan92	Unclassified A	DNA-binding v	1.0	122-126	RRARS
FeP2	Pigeonpox vir	DNA-binding v	1.0	122-126	RRARS

Strain Name	Species Name	Protein Name	Score	Range	Matched Sequence
AHRV241013	Atlantic halibu	RNA-depende	1.0	530-534	RRARS
1001	Micropterus sa	VP2	1.0	529-533	RRARS
AHRV060513	Atlantic halibu	RNA-depende	1.0	534-538	RRARS
AHRV241013	Atlantic halibu	RNA-depende	1.0	534-538	RRARS
UNKNOWN-H	Aquareovirus	VP2	1.0	529-533	RRARS
UNKNOWN-N	Mycoreovirus	hypothetical p	1.0	962-966	RRARS
UNKNOWN-A	Mycoreovirus	hypothetical p	1.0	962-966	RRARS
UNKNOWN-N	Aquareovirus	putative viral p	1.0	495-499	RRARS
UNKNOWN-N	Aquareovirus	VP2	1.0	529-533	RRARS
GSH1	Fall chinook a	VP2	1.0	530-534	RRARS
San Marcos 2	Etheostoma fo	RNA-depende	1.0	529-533	RRARS
AGCRV_PB0	Aquareovirus	VP2	1.0	529-533	RRARS
MERV-1	Unclassified A	VP2	1.0	530-534	RRARS
Golden shiner	Aquareovirus	RNA-depende	1.0	529-533	RRARS
GSH1	Fall chinook a	VP2	1.0	530-534	RRARS
San Marcos 2	Etheostoma fo	RNA-depende	1.0	529-533	RRARS
GZ1208	Aquareovirus	RNA-depende	1.0	529-533	RRARS
AGCRV_PB0	Aquareovirus	VP2	1.0	529-533	RRARS
UNKNOWN-E	Aquareovirus	RNA-depende	1.0	529-533	RRARS
UNKNOWN-A	Aquareovirus	RNA-depende	1.0	67-71	RRARS
UNKNOWN-A	Aquareovirus	RNA-depende	1.0	291-295	RRARS
UNKNOWN-A	Aquareovirus	RNA-depende	1.0	529-533	RRARS
UNKNOWN-A	Aquareovirus	putative viral p	1.0	495-499	RRARS
873	Aquareovirus	inner capsid p	1.0	529-533	RRARS
UNKNOWN-A	Aquareovirus	VP2	1.0	529-533	RRARS

Appendix VI

Codes used in this project

Code to create batches of FASTA files with 500 sequences each:

```
import xlrd
import os.path
os.chdir(r"C:\Users\svpan\Downloads\All_seqs")
wb = xlrd.open_workbook('all fusions.xlsx')
wb.sheet_names()
sh = wb.sheet_by_index(0)
a=0
list1=[]
while a<=57000:
   list1.append(a)
   a+=500
for jk in list1:
   i = 0
   jk2=jk+500
   name1="seqs_"+str(jk2)+".fasta"

   print(jk)
   print(jk2)
   with open(name1, "a") as my_file:
      for i in range(jk,jk2):
         sequence = sh.cell(i,17).value
         nterminus = sh.cell(i,8).value
         cterminus = sh.cell(i,12).value
         if sequence:
            DB1 = (">"+ nterminus + "/" + cterminus + "\n" + sequence)
            my_file.write(DB1 + '\n')
         i = i+1
```

To remove duplicates:

```
import xlrd
import xlsxwriter
import re
from itertools import takewhile
import os
os.chdir(r"C:\Users\svpan\Downloads\LINKER_DB" )
```

```
number=0
numos=24
dtf=str(numos)+'_LINKERS'
prot=[]
seq=[]
linkl=[]
linker=[]
nt=[]
ct=[]
def column_len(sheet, index):
    col_values = sheet.col_values(index)
    col_len = len(col_values)
    for _ in takewhile(lambda x: not x, reversed(col_values)):
        col_len -= 1
    return col_len

name1=str(dtf)+".xlsx"
wb = xlrd.open_workbook(name1)
sheet = wb.sheet_by_index(number)


for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    prot.append(row[0].value)
    seq.append(row[1].value)
    linkl.append(row[2].value)
    nt.append(row[3].value)
    ct.append(row[4].value)
    linker.append(row[5].value)



total=[]
count=0
for i in prot:

string=prot[count]+"**"+seq[count]+"**"+str(linkl[count])+"**"+str(nt[count])+"**"+str(ct[cou
nt])+"**"+str(linker[count])
    total.append(string)
```

```
    count+=1
for a in total:
    dfnew = a.split("**")
    print(dfnew)
removed=[]
removed = list(dict.fromkeys(total))
removed.pop(0)
dfnew=[]
protn=[]
seqn=[]
linkln=[]
linkern=[]
ntn=[]
ctn=[]

for z in removed:
    dfnew = z.split("**")
    protn.append(dfnew[0])
    seqn.append(dfnew[1])
    linkln.append(dfnew[2])
    ntn.append(dfnew[3])
    ctn.append(dfnew[4])
    linkern.append(dfnew[5])



import pandas as pd

df=pd.DataFrame({"Protein":pd.Series(protn), "Sequence":pd.Series(seqn),"Linker
Length":pd.Series(linkln),"N term":pd.Series(ntn),"C
term":pd.Series(ctn),"Linker":pd.Series(linkern)})

df.to_excel('link'+dtf+'dup.xlsx', index=False)
```

Program to parse BLAST xml output:

```
#parse the xml file and store it
import os
import pandas as pd
```

```
def getseqs(num):
    print("here")
    import xlsxwriter
    from Bio.Blast import NCBIXML

    import pandas as pd

    i=num
    f=i-1
    name1="res_"+str(i)+".xml"
    name2="seqs_"+str(f)+".txt"
    name3=str(i)+"_dfand"+".xlsx"


    workbook = xlsxwriter.Workbook(name3)
    worksheet = workbook.add_worksheet()

    #open the results file
    result=open(name1,"r")
    records= NCBIXML.parse(result)
    item=records

    trial_3=pd.DataFrame()

    alignments=[]
    start=[]
    end=[]
    linker=[]
    matched_indexes=[]
    qid=[]
    allq=[]
    defn=[]

    seqlist=[]
    s=[]
    f=open(name2, "r")
    ksj= 1
    for line in f.readlines():
```

```
        if ksj % 2 == 0 :
            seqlist.append(line.rstrip())
        ksj += 1

    #list1.pop(0)
    ks=0
    trialdf=pd.DataFrame()
    trial_2=pd.DataFrame()
    for record in records:

        allq.append(record.query)
        for alignment in record.alignments:
            for hsp in alignment.hsps:
                if hsp.expect < 0.5:
                    start=[]
                #only first hsp is taken for each hit, this is the hsp with the best score
                    if alignment.hit_def:
                        start.append(seqlist[ks])

                    start.append(hsp.query_start)
                    start.append(hsp.query_end)
                    start.append(alignment.hit_def)
                    allq.append('')
                    #trialdf=pd.DataFrame({"Sequence":pd.Series(start[0]),
"Start":pd.Series(start[1]),"End":pd.Series(start[2]),"Alignment Definition":pd.Series(start[3])})
                    trialdf=pd.DataFrame({"Sequence":pd.Series(start[0]),
"Start":pd.Series(start[1]),"End":pd.Series(start[2]),"Alignment Definition":pd.Series(start[3])})
                    trial_2=pd.concat([trial_2,trialdf], axis=0)

                    break

        ks=ks+1
        #print(allq)
        allq.pop()

    newl=[]
    cterm=[]
    nterm=[]
    for nj in allq:
```

```
        if nj==":
            cterm.append(")
            nterm.append(")
        else:
            #print(newl)
            newl=nj.split('/')
            cterm.append(newl[1])
            nterm.append(newl[0])



    trial_2.insert(0,"pr1",nterm)
    trial_2.insert(1,"pr2",cterm)
    finaldf=trial_2.append({"pr1":"END", "pr2":' ',"Sequence":' ',"Start":' ',"End":' ',"Alignment
Definition":' '}, ignore_index=True)
    finaldf.to_excel(name3, index=False)


os.chdir(r"C:\Users\svpan\Downloads\parse" )
path=os.getcwd()
files = os.listdir(path)
seqnum=[]
for i in files:
    if (i == "all fusions.xlsx") or (i.endswith('.xml')) or (i.endswith('.txt')):
        continue
    else:
        string1=i.split("_")
        string2=string1[1].split(".")
        num=int(string2[0])

        getseqs(num)
```

Clean parsed file:

```
#delete the first row
from Bio.Blast import NCBIXML
import pandas as pd

import openpyxl
```

```
os.chdir(r"C:\Users\svpan\Downloads\WRONG" )
path=os.getcwd()
files = os.listdir(path)
seqnum=[]
for i in files:

    if (i.endswith('dfand.xlsx')):
        filename = i
        wb = openpyxl.load_workbook(filename)
        sheet = wb['Sheet1']
        status = sheet.cell(sheet.min_row, 1).value
        print(status)
        sheet.delete_rows(sheet.min_row, 1)
        wb.save(filename)
```

Get names not matched by dictionary:

```
import pandas as pd
import xlrd
import xlsxwriter
import re

df = pd.read_excel('red_link.xlsx') # can also index sheet by name or fetch all sheets
pr1 = df['pr1'].tolist()
pr2 = df['pr2'].tolist()



excel_file = 'gene_dictionary.xlsx'
book = xlrd.open_workbook(excel_file)
sheet = book.sheet_by_index(0)
gene_names=[]
for row in range(sheet.nrows):
        gene_names.append(sheet.cell(row,1).value)

main_list = list(set(pr2) - set(gene_names))
main_list2 = list(set(pr1) - set(gene_names))

print(main_list)
```

```
print(main_list2)
```

Check file against gene dictionary:

```
import pandas as pd

df = pd.read_excel('remove dups.xlsx') # can also index sheet by name or fetch all sheets
mylist = df['pr1'].tolist()

df2=pd.read_excel('50k.xlsx')

complist=df2['misc'].tolist()

mylist = list(dict.fromkeys(mylist))

for element in mylist:
    if element not in complist:
        print(element)
```

Code to help create gene dictionary:

```
import pandas as pd

df = pd.read_excel('all fusions.xlsx') # can also index sheet by name or fetch all sheets
nterminus = df['nterm'].tolist()
cterminus = df['cterm'].tolist()

s = set(nterminus)
nt = list(s)
s1 = set(cterminus)
ct = list(s1)

df_nc=pd.DataFrame({'Nterminus': pd.Series(nt), 'Cterminus': pd.Series(ct)})
df_nc.to_excel('Proteins_fordict.xlsx', index=False)
```

Parse result file to generate input file for linker program:

```
import xlsxwriter
from Bio.Blast import NCBIXML
```

```
import pandas as pd

i=14000
f=i-1
name1="res_"+str(i)+".xml"
name2="seqs_"+str(f)+".txt"
name3=str(i)+"_dfand"+".xlsx"


workbook = xlsxwriter.Workbook(name3)
worksheet = workbook.add_worksheet()

#open the results file
result=open(name1,"r")
records= NCBIXML.parse(result)
item=records

trial_3=pd.DataFrame()

alignments=[]
start=[]
end=[]
linker=[]
matched_indexes=[]
qid=[]
allq=[]
defn=[]

seqlist=[]
s=[]
f=open(name2, "r")
ksj= 1
for line in f.readlines():
    if ksj % 2 == 0 :
        seqlist.append(line.rstrip())
    ksj += 1

#list1.pop(0)
ks=0
```

```
trialdf=pd.DataFrame()
trial_2=pd.DataFrame()
for record in records:

    allq.append(record.query)
    for alignment in record.alignments:
        for hsp in alignment.hsps:
            start=[]
            #only first hsp is taken for each hit, this is the hsp with the best score
            if alignment.hit_def:
                start.append(seqlist[ks])

            start.append(hsp.query_start)
            start.append(hsp.query_end)
            start.append(alignment.hit_def)
            allq.append('')
            #trialdf=pd.DataFrame({"Sequence":pd.Series(start[0]),
"Start":pd.Series(start[1]),"End":pd.Series(start[2]),"Alignment Definition":pd.Series(start[3])})
            trialdf=pd.DataFrame({"Sequence":pd.Series(start[0]),
"Start":pd.Series(start[1]),"End":pd.Series(start[2]),"Alignment Definition":pd.Series(start[3])})
            trial_2=pd.concat([trial_2,trialdf], axis=0)

            break
    ks=ks+1
    #print(allq)
    allq.pop()

newl=[]
cterm=[]
nterm=[]
for nj in allq:
    if nj=='':
        cterm.append('')
        nterm.append('')
    else:
        #print(newl)
        newl=nj.split('/')
        cterm.append(newl[1])
        nterm.append(newl[0])
```

```
trial_2.insert(0,"pr1",nterm)
trial_2.insert(1,"pr2",cterm)
finaldf=trial_2.append({"pr1":"END", "pr2":' ',"Sequence":' ',"Start":' ',"End":' ',"Alignment
Definition":' '}, ignore_index=True)
finaldf.to_excel(name3, index=False)
```

Program to get Linker regions:

```
#gcf is the number of alignments so in this case we took 100 alignments for all results
gcf=100
indcnt=0
#strRow=0
x=[]
cnt=0
import pandas as pd
import os
df_all=pd.DataFrame()
newdf=pd.DataFrame()
secdf=pd.DataFrame()
os.chdir(r"C:\Users\svpan\Downloads\parse" )
path=os.getcwd()

dtf=21000

def getfinallist():
    import xlrd
    import xlsxwriter
    import re
    from itertools import takewhile

    ntermprtn=[]
    indexPosList = []

    def column_len(sheet, index):
        col_values = sheet.col_values(index)
        col_len = len(col_values)
```

```
        for _ in takewhile(lambda x: not x, reversed(col_values)):
            col_len -= 1
        return col_len

    name1=str(dtf)+"_dfand"+".xlsx"
    wb2 = xlrd.open_workbook(name1)
    sheet = wb2.sheet_by_index(0)


    for i in range(0, sheet.nrows):
        row = sheet.row_slice(i)
        ntermprtn.append(row[0].value)

    k=0

    for i in ntermprtn:
        k=k+1
        if i:
            n=k
            indexPosList.append(n)

    return indexPosList

list1=getfinallist()
#print(list1)


def ncterminal(index1,index_sec,cnt,dtf):

    import xlrd
    import xlsxwriter
    import re
    from Bio.Blast import NCBIXML
    #To store the c and n terminal of all the fusion proteins in 2 lists
    ntermprtn=[]
    ctermprtn=[]
    start=[]
    end=[]
    defn=[]
```

```
indexPosList = []
seqence=[]

name1=str(dtf)+"_dfand"+".xlsx"

#get the parsed data
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(0)

for i in range(index1-1,index_sec-1):
   ntermprtn.append(sheet.cell_value(i,0))
for i in range(index1-1,index_sec-1):
   ctermprtn.append(sheet.cell_value(i,1))
for i in range(index1-1,index_sec-1):
   seqence.append(sheet.cell_value(i,2))
for i in range(index1-1,index_sec-1):
   start.append(sheet.cell_value(i,3))
for i in range(index1-1,index_sec-1):
   end.append(sheet.cell_value(i,4))
for i in range(index1-1,index_sec-1):
   defn.append(sheet.cell_value(i,5))



print(ntermprtn)
#will have to loop for multiple proteins
patc=ctermprtn[0]

#creating dictionary to understand the alignment definition line
excel_file = 'gene_dictionary.xlsx'
book = xlrd.open_workbook(excel_file)
sheet = book.sheet_by_index(0)
ensembl1=0
gene1=1
gene_names1=2
gene_desc1 = 3
uniprot1 = 4
gene_syn1=5

gene_syn=[]
```

```
gene_names=[]
gene_desc=[]
uniprot=[]
ensembl=[]
gene=[]

for row in range(sheet.nrows):
    gene_syn.append(sheet.cell(row,gene_syn1).value)
    gene_desc.append(sheet.cell(row,gene_desc1).value)
    gene_names.append(sheet.cell(row,gene_names1).value)
    uniprot.append(sheet.cell(row,uniprot1).value)
    ensembl.append(sheet.cell(row,ensembl1).value)
    gene.append(sheet.cell(row,gene1).value)

gsc=[]
gdc=[]
upc=[]
enc=[]
gname=[]
gec=[s for s in gene if patc in s]



if gec != []:
    element=gec[0]
    indexdict=gene.index(element)
else:
    print('Error: Dictionary does not have C terminal protein: '+ patc)


gdc=gene_desc[indexdict]
gdc=gdc.split(",")
gsc=gene_syn[indexdict]
gsc=gsc.split(",")
upc.append(uniprot[indexdict])
enc=ensembl[indexdict]
enc=enc.split(";")
enc=list(filter(None, enc))
gname=gene_names[indexdict]
```

```
#print(gname)
gname=gname.split(",")


i=0


matchc=[]
matchc_al=[]

h=0

for i in defn:
    titlestr=defn[h]
    if gsc != []:
        if titlestr.find(gsc[0]) != -1:
            matchc.append(defn.index(titlestr))

    if gdc != []:
        if titlestr.find(gdc[0]) != -1:
            matchc.append(defn.index(titlestr))

    if upc != []:
        if titlestr.find(upc[0]) != -1:
            matchc.append(defn.index(titlestr))

    if enc != []:
        if titlestr.find(enc[0]) != -1:
            matchc.append(defn.index(titlestr))

    if gec != []:
        if titlestr.find(gec[0]) != -1:
            matchc.append(defn.index(titlestr))

    if gname != []:
        if titlestr.find(gname[0]) != -1:
            matchc.append(defn.index(titlestr))
```

```
    h=h+1

#finalc = [s for s in defn if pat in s]
#To get the hit definition and match it for homo sapiens and the n terminal protein
#patn="none"

patn=ntermprtn[0] #will have to loop for multiple proteins
#matchhsn = [s for s in defnn if "Homo sapiens" in s]
#finaln = [s for s in defn if pat in s]



gsn=[]
gdn=[]
upn=[]
enn=[]
gnamen=[]
gen=[s for s in gene if patn in s]



if gen != []:
    element=gen[0]

    indexdictn=gene.index(element)


else:
    print('Error: The dictionary does not contain N terminal protein:'+ patn)




gdn.append(gene_desc[indexdictn])
gsn=gene_syn[indexdictn]
gsn=gsn.split(",")
upn.append(uniprot[indexdictn])
enn=ensembl[indexdictn]
```

```
enn=enn.split(";")
enn=list(filter(None, enn))
gnamen=gene_names[indexdictn]
gnamen=gnamen.split(",")


matchn=[]
matchn_al=[]

h=0

for i in defn:
    titlestr=defn[h]
    if gsn != []:
        if titlestr.find(gsn[0]) != -1:
            indn=defn.index(titlestr)
            matchn.append(indn)
    if gdn != []:
        if titlestr.find(gdn[0]) != -1:
            indn=defn.index(titlestr)
            matchn.append(indn)
    if upn != []:
        if titlestr.find(upn[0]) != -1:
            indn=defn.index(titlestr)
            matchn.append(indn)
    if enn != []:
        if titlestr.find(enn[0]) != -1:
            indn=defn.index(titlestr)
            matchn.append(indn)
    if gen != []:
        if titlestr.find(gen[0]) != -1:
            indn=defn.index(titlestr)
            matchn.append(indn)
    if gnamen != []:
        if titlestr.find(gnamen[0]) != -1:
            indn=defn.index(titlestr)
            matchn.append(indn)

    h=h+1
```

```
#print('This is matchn:')
#print(matchn)

#store the indices of the matches in a list


end_match_c=[]
end_match_n=[]
start_match_n=[]
start_match_c=[]
im=0
ik=0

while ik < len(matchc):
   end_match_c.append(end[matchc[ik]])
   ik=ik+1

while im < len(matchn):
   end_match_n.append(end[matchn[im]])
   im=im+1
ik=0
im=0
while ik < len(matchc):
   start_match_c.append(start[matchc[ik]])
   ik=ik+1
while im < len(matchn):
   start_match_n.append(start[matchn[im]])
   im=im+1

import numpy as np
import pandas as pd

fusprtn=[]
fusprtn.append(patn+"/"+patc)
sq=seqence[0]
```

```
    df = pd.DataFrame({'Protein': pd.Series(fusprtn),'Sequence': pd.Series(sq), 'Start N':
pd.Series(start_match_n), 'End N': pd.Series(end_match_n), 'Start C': pd.Series(start_match_c),
'End C': pd.Series(end_match_c)})

    return indexPosList, df, end_match_n, start_match_c,fusprtn,sq



def get_linker(fp,sq,mn,en,st):
    enn=en
    stc=st
    seq = sq
    prt = fp
    print("PROTEIN")
    print(fp)
    print("ENNNNNNNNNNNNNNNNNNNNNNNNNN")
    print(enn)
    print("STCCCCCCCCCCCCCCCCCCCCCCCCCC")
    print(stc)
    mx1=len(enn)
    mx2=len(stc)
    linklistlen=[]
    linklist=[]
    nloc=[]
    cloc=[]
    if mx1==0 or mx2==0:
        nloc.append('none')
        cloc.append('none')
        linklistlen.append('no linker region found')
    if mx1<mx2:
        sm=enn
        bh=stc

    else:
        sm=stc
        bh=enn

    jm=0
    jk=0
```

```
#not sure if third condition is okay check with maam
if sm == enn:
    for jm in sm:
        for jk in bh:
            if jm>jk or jm==jk:
                nloc.append('none')
                cloc.append('none')
                linklistlen.append('no linker region found')
            elif (jk-jm) == 1:
                nloc.append('none')
                cloc.append('none')
                linklistlen.append('no linker region found')
            elif (jk-jm) >= 90:
                nloc.append('none')
                cloc.append('none')
                linklistlen.append('no linker region found')
            else:
                nloc.append(jm)
                print("NLOC++++++++++++++++++++++++")
                print(nloc)
                cloc.append(jk)
                print("CLOC++++++++++++++++++++++++")
                print(cloc)
                linklistlen.append(jk-jm)

else:
    for jm in sm:
        for jk in bh:
            if jm<jk or jm==jk:
                nloc.append('none')
                cloc.append('none')
                linklistlen.append('no linker region found')
            elif (jm-jk) == 1:
                nloc.append('none')
                cloc.append('none')
                linklistlen.append('no linker region found')
            elif (jm-jk) >= 90:
                nloc.append('none')
                cloc.append('none')
```

```
                linklistlen.append('no linker region found')
            else:
                nloc.append(jk)
                cloc.append(jm)
                print("NLOC++++++++++++++++++++++++")
                print(nloc)
                print("CLOC++++++++++++++++++++++++")
                print(cloc)
                linklistlen.append(jm-jk)

    a=0
    b=0
    lin=[]
    nind=0

    for a in nloc:
        if a=='none' or cloc[nind]=='none':
            lin.append('no linker')

        else:
            lin.append(sq[int(a):int(cloc[nind])])

        nind+=1


    df2=pd.DataFrame({'Protein': pd.Series(fp),'Sequence': pd.Series(seq),'Linker Length':
pd.Series(linklistlen), 'N term': pd.Series(nloc), 'C term': pd.Series(cloc), 'Linker':
pd.Series(lin)})

    return df2




#print(seqlist)
#list1.pop(0)

mn=1
s=[]
```

```
inde1=0
index1=0
inde2=0
index2=0

for i in list1:
    if i == list1[len(list1)-1]:
        break
    #print(i)
    inde1=list1.index(i)
    index1=list1[inde1]

    #print(index1)
    inde2=inde1+1

    index_sec=list1[inde2]


    (pos,l,strpt,endpt,fp,sq) = ncterminal(index1,index_sec,cnt,dtf)

    df_all= pd.concat([df_all], axis=0)
    newdf=pd.concat([newdf,l], axis=0)
    cnt=cnt+1
    (df_2)=get_linker(fp,sq,mn,strpt,endpt)

    secdf=pd.concat([secdf,df_2], axis=0)
    mn=mn+1



print(list1)

name2="res_"+str(dtf)+".xlsx"
name3="linker_"+str(dtf)+".xlsx"
newdf.to_excel(name2, index=False)
secdf.to_excel(name3, index=False)
```

Checking for missing proteins after cleaning:

```
import pandas as pd
```

```
excel_data_df = pd.read_excel('All_linkers.xlsx')
excel_data_df2 = pd.read_excel('finalall.xlsx')
prot=excel_data_df['Protein'].tolist()
prot2=excel_data_df['Protein'].tolist()

unique_prot = list(dict.fromkeys(prot))
unique_prot2 = list(dict.fromkeys(prot2))
print(unique_prot)
print(unique_prot2)
common=list(set(unique_prot).intersection(unique_prot2))
one_not_two = set(unique_prot).difference(unique_prot2)
print(one_not_two)
two_not_one = set(unique_prot2).difference(unique_prot)
print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++")
print(two_not_one)
```

Concatenate all final results into one excel file:

```
import os
import pandas as pd
os.chdir(r"C:\Users\svpan\Downloads\LINKER_DB" )
path=os.getcwd()
files = os.listdir(path)
df = pd.DataFrame()
print(len(files))
for f in files:
    data = pd.read_excel(f, 'Sheet1')
    df = df.append(data)

df.to_excel("all_linkers.xlsx", index=False)
```

Concatenate results files into excel documents:

```
import os
import pandas as pd
os.chdir(r"C:\Users\svpan\Downloads\linkers" )
```

```
path=os.getcwd()
files = os.listdir(path)
df = pd.DataFrame()
print(len(files))
df2 = pd.DataFrame()
df3 = pd.DataFrame()
df4= pd.DataFrame()
for f in files:
    data = pd.read_excel(f, 'Sheet1')
    if files.index(f) < 30:
        df = df.append(data)
        continue
    if files.index(f) < 50:
        df2 = df2.append(data)
        continue
    if files.index(f) < 90:
        df3 = df3.append(data)
        continue
    if files.index(f) < 120:
        df4 = df4.append(data)
        continue

df.to_excel("1LINKERS.xlsx", index=False)
df2.to_excel("2LINKERS.xlsx", index=False)
df3.to_excel("3LINKERS.xlsx", index=False)
df4.to_excel("4LINKERS.xlsx", index=False)
```

To add titles and sequences:

```
import pandas as pd
#to add titles and sequences
dtf="3LINKERS"

import xlrd
import xlsxwriter
import re
from itertools import takewhile
```

```
ntermprtn=[]
indexPosList = []

def column_len(sheet, index):
    col_values = sheet.col_values(index)
    col_len = len(col_values)
    for _ in takewhile(lambda x: not x, reversed(col_values)):
        col_len -= 1
    return col_len

name1=str(dtf)+".xlsx"
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(0)

excel_data_df = pd.read_excel(name1)
prot=excel_data_df['Protein'].tolist()
seq=excel_data_df['Sequence'].tolist()
llen=excel_data_df['Linker Length'].tolist()
nt=excel_data_df['N term'].tolist()
ct=excel_data_df['C term'].tolist()
linker=excel_data_df['Linker'].tolist()
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    ntermprtn.append(row[0].value)

newprot=[]
for i in prot:
    res = isinstance(i, str)
    ind=prot.index(i)
    if res == True:
        newprot.append(i)
        prev=i
    else:
        newprot.append(prev)
newseq=[]

for j in seq:
    sres = isinstance(j, str)
    sind=seq.index(j)
```

```
    if sres == True:
        newseq.append(j)
        prevs=j
    else:
        newseq.append(prevs)

df=pd.DataFrame({"Protein":pd.Series(newprot), "Sequence":pd.Series(newseq),"Linker
Length":pd.Series(llen),"N term":pd.Series(nt),"C
term":pd.Series(ct),"Linker":pd.Series(linker)})

df.to_excel("23_LINKERS.xlsx", index=False)
```

Code to convert excel file to csv:

```
import pandas as pd

read_file = pd.read_excel ('Linker file with rem.xlsx')
read_file.to_csv ('No_nolinker.csv', index = None, header=True)
```

Code to find MEROPS table element:

```
import requests
import lxml.html as lh
import pandas as pd

url='https://www.ebi.ac.uk/merops/cgi-bin/pepsum?id=S01.021;type=P'
#Create a handle, page, to handle the contents of the website
page = requests.get(url)
#Store the contents of the website under doc
doc = lh.fromstring(page.content)
#Parse data that are stored between <tr>..</tr> of HTML
tr_elements = doc.xpath('//tr')

[len(T) for T in tr_elements[:12]]
print(tr_elements)
tr_elements = doc.xpath('//tr')
#Create empty list
col=[]
i=0
#For each row, store each first element (header) and an empty list
for t in tr_elements[17]:
```

```
    i+=1
    name=t.text_content()
    #print '%d   %s'%(i,name)
    col.append((name,[]))

for j in range(1,len(tr_elements)):
    #T is our j'th row
    T=tr_elements[j]

    #If row is not of size 10, the //tr data is not from our table
    if len(T)!=10:
        break

    #i is the index of our column
    i=0

    #Iterate through each element of the row
    for t in T.iterchildren():
        data=t.text_content()
        #Check if row is empty
        if i>0:
        #Convert any numerical value to integers
            try:
                data=int(data)
            except:
                pass
        #Append the data to the empty list of the i'th column
        col[i][1].append(data)
        #Increment i for the next column
        i+=1

[len(C) for (title,C) in col]
Dict={title:column for (title,column) in col}
df=pd.DataFrame(Dict)
df.to_excel('MEROPSTABLE.xlsx', index=False)
```

Code to retrieve table:

```
import pandas as pd
import xlrd
```

```
import xlsxwriter
import re
from itertools import takewhile
dtf='MEROPS_proteases'
merops_id=[]
def column_len(sheet, index):
    col_values = sheet.col_values(index)
    col_len = len(col_values)
    for _ in takewhile(lambda x: not x, reversed(col_values)):
        col_len -= 1
    return col_len

name1=str(dtf)+".xlsx"
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(1)

prot=[]
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    merops_id.append(row[0].value)
    prot.append(row[1].value)
merops_id.pop(0)
prot.pop(0)
table2 = pd.DataFrame()
count=500
for i in merops_id:
    if count == 536:
            break
    print(merops_id[count])
    html_url="https://www.ebi.ac.uk/merops/cgi-bin/pepsum?id="+str(merops_id[count])
    print(html_url)
    tables = pd.read_html(html_url)
    length=len(tables)
    if length <= 5:
        count+=1
        continue
    else:
        print("HERE")
        table=pd.DataFrame(tables[5])
```

```
        table2=table2.append({'id':merops_id[count], 'prot':prot[count]}, ignore_index=True)
        table2=table2.append(table, ignore_index=True)
        count+=1

table2.to_excel("table537.xlsx", index=False)
```

Code to clean data retrieved from MEROPS:

```
import pandas as pd
import numpy as np
import Bio

df = pd.read_excel('MEROPS_sites.xlsx')

d = {'CYS': 'C', 'ASP': 'D', 'SER': 'S', 'GLN': 'Q', 'LYS': 'K',
     'ILE': 'I', 'PRO': 'P', 'THR': 'T', 'PHE': 'F', 'ASN': 'N',
     'GLY': 'G', 'HIS': 'H', 'LEU': 'L', 'ARG': 'R', 'TRP': 'W',
     'ALA': 'A', 'VAL':'V', 'GLU': 'E', 'TYR': 'Y', 'MET': 'M'}
letter3=list(d.keys())
df.dropna()
pept=[]
peptn=[]
pept=df['PEPTIDASE']
subs=df['Substrate']
uniprot=df['Uniprot']
Residuerng=df['Residue range']
P1=df['P1']
P2=df['P2']
P3=df['P3']
P4=df['P4']
P1pr=df['P1prime']
P1n=[]
P2n=[]
P3n=[]
P4n=[]
P1prime=[]
skip=[]
tripeptide=[]
tetrapeptide=[]
```

```
for k in pept:
    k=str(k)
    peptn.append(k)



for i in P1:
    i=str(i)
    i=i.upper()
    if i in letter3:
        #print('here')
        #print(d[i])
        P1n.append(d[i])
    else:
        P1n.append('none')
for m in P2:
    m=str(m)
    m=m.upper()
    if m in letter3:
        P2n.append(d[m])
    else:
        P2n.append('none')
for n in P3:
    n=str(n)
    n=n.upper()
    if n in letter3:
        P3n.append(d[n])
    else:
        P3n.append('none')
for o in P4:
    o=str(o)
    o=o.upper()
    if o in letter3:
        P4n.append(d[o])
    else:
        P4n.append('none')
for f in P1pr:
    f=str(f)
    f=f.upper()
```

```
    if f in letter3:
        P1prime.append(d[f])
    else:
        P1prime.append('none')

count=0
tetra=[]
tri=[]
pri=[]
for q in P1n:
    string4=P4n[count]+P3n[count]+P2n[count]+P1n[count]
    string3=P3n[count]+P2n[count]+P1n[count]
    stringpr=P3n[count]+P2n[count]+P1n[count]+"**"+str(P1prime[count])
    tetra.append(string4)
    tri.append(string3)
    pri.append(stringpr)
    count+=1
P1prm=[]
for z in tri:
    if 'none' in z:
        tri[tri.index(z)]='none'



for p in tetra:
    if 'none' in p:
        tetra[tetra.index(p)]='none'

#for a in pri:
 #   fnew=a.split("**")
  #  pri[pri.index(a)]=fnew[1]



df2=pd.DataFrame({'Peptidase': pd.Series(peptn),'Tripeptide': pd.Series(tri),'Tetrapeptide+P1
prime': pd.Series(tetra)})

df2.to_excel('res_merops_tetraser.xlsx', index=False)
```

Code to retrieve tetrapeptides found in linker sequence:

```
#tetrapeptides
import xlrd
import xlsxwriter
import re
from itertools import takewhile
from string import printable
import os
os.chdir(r"C:\Users\svpan\Downloads\LINKER_TETRA" )
path=os.getcwd()
number=0
dtf='Remove redundant linkers.xlsx'

name1=str(dtf)
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(0)

ind=[]
sequence=[]
llen=[]
protein1=[]
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    protein1.append(row[0].value)
    sequence.append(row[5].value)
res=[]
df2=pd.DataFrame()
for a in sequence:
    if a.find("_") == 1:
        index190=sequence.index(a)
        res.append("invalid character")
        sequence.pop(index190)
        protein1.pop(index190)
        continue
    else:
        res.append("valid")
indexseq=0
for seq in sequence:
    if indexseq <= 500:
        indexseq=sequence.index(seq)
```

```
proteinseq=protein1[indexseq]
site1=[]
indexPosList = []
found=[]
fi=[]
prot=[]
foundp=[]
dtf='res_merops_P1pr'
def column_len(sheet, index):
    col_values = sheet.col_values(index)
    col_len = len(col_values)
    for _ in takewhile(lambda x: not x, reversed(col_values)):
        col_len -= 1
    return col_len

name1=str(dtf)+".xlsx"
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(number)

prime=[]
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    site1.append(row[2].value)
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    prot.append(row[0].value)


#print(site1)
for w in site1:
    if w in seq:
        fi.append([i for i, x in enumerate(site1) if x == w])


unique_data = [list(x) for x in set(tuple(x) for x in fi)]
print(unique_data)
p1prime=[]
for n in unique_data:
    for j in n:
```

```
            foundp.append(prot[j])
            found.append(site1[j])

        z=0
        #print(prot)
        ind=[]
        lastind=[]
        print(found)
        for i in found:
           ind.append(seq.find(i)+1)
        K=3
        lastind = [x + K for x in ind]



        finind=[]
        for i in found:
           index=seq.find(i)

           finind.append(seq[index+3])



        import pandas as pd
        df=pd.DataFrame({'Protein': pd.Series(foundp),'Site': pd.Series(found),'Start':
pd.Series(ind),'End': pd.Series(lastind),'Fusion protein':pd.Series(proteinseq),'Linker
sequence':pd.Series(seq)})
        df2=pd.concat([df2,df], axis=0)
        print(df)
        proteinseq=proteinseq.split("/")
        if proteinseq[0] == 'Protein':
           continue



df2.to_excel('Linkers_tetra_500.xlsx', index=False)
```

Code to retrieve tripeptides from linker sequence:

```
#tripeptides
import xlrd
import xlsxwriter
```

```
import re
from itertools import takewhile
from string import printable
import os
os.chdir(r"C:\Users\svpan\Downloads\LINKER_TRI" )
path=os.getcwd()
number=0
dtf='Remove redundant linkers.xlsx'

name1=str(dtf)
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(0)

ind=[]
sequence=[]
llen=[]
protein1=[]
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    protein1.append(row[0].value)
    sequence.append(row[5].value)
res=[]
for a in sequence:
    if a.find("_") == 1:
        index190=sequence.index(a)
        res.append("invalid character")
        sequence.pop(index190)
        protein1.pop(index190)
        continue
    else:
        res.append("valid")
cnt=0
for seq in sequence:
    indexseq=sequence.index(seq)
    proteinseq=protein1[indexseq]
    site1=[]
    indexPosList = []
    found=[]
    fi=[]
```

```
prot=[]
foundp=[]
dtf='res_merops_P1pr'
def column_len(sheet, index):
    col_values = sheet.col_values(index)
    col_len = len(col_values)
    for _ in takewhile(lambda x: not x, reversed(col_values)):
        col_len -= 1
    return col_len

name1=str(dtf)+".xlsx"
wb2 = xlrd.open_workbook(name1)
sheet = wb2.sheet_by_index(number)

prime=[]
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    site1.append(row[1].value)
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    prot.append(row[0].value)



#print(site1)
for w in site1:
    if w in seq:
        fi.append([i for i, x in enumerate(site1) if x == w])


unique_data = [list(x) for x in set(tuple(x) for x in fi)]
print(unique_data)
p1prime=[]
for n in unique_data:
    for j in n:
        foundp.append(prot[j])
        found.append(site1[j])

z=0
#print(prot)
```

```
    ind=[]
    lastind=[]
    print(found)
    for i in found:
        ind.append(seq.find(i)+1)
    K=2
    lastind = [x + K for x in ind]


    finind=[]
    for i in found:
        index=seq.find(i)

        finind.append(seq[index+3])


    import pandas as pd
    df=pd.DataFrame({'Protein': pd.Series(foundp),'Site': pd.Series(found),'Start':
pd.Series(ind),'End': pd.Series(lastind)})
    proteinseq=proteinseq.split("/")
    if proteinseq[0] == 'Protein':
        continue
    print(proteinseq)
    name7=proteinseq[0]+"_"+proteinseq[1]
    df.to_excel(str(name7)+'(tri'+str(cnt)+').xlsx', index=False)
    cnt=cnt+1
```

Code to calculate amino acid counts:

```
import xlrd
import xlsxwriter
import re
from itertools import takewhile
from string import printable
import os
os.chdir(r"C:\Users\svpan\Downloads\thesis res" )
d = {'CYS': 'C', 'ASP': 'D', 'SER': 'S', 'GLN': 'Q', 'LYS': 'K',
     'ILE': 'I', 'PRO': 'P', 'THR': 'T', 'PHE': 'F', 'ASN': 'N',
```

```
    'GLY': 'G', 'HIS': 'H', 'LEU': 'L', 'ARG': 'R', 'TRP': 'W',
    'ALA': 'A', 'VAL':'V', 'GLU': 'E', 'TYR': 'Y', 'MET': 'M'}
letter3=list(d.values())
print(letter3)
wb2 = xlrd.open_workbook("red_link.xlsx")
sheet = wb2.sheet_by_index(0)

linker=[]

for i in range(0, sheet.nrows):
   row = sheet.row_slice(i)
   linker.append(row[1].value)
res=[]
cnt=0
Amino=[]
counts=[]
def count(s, c) :

   # Count variable
   res = 0

   for i in range(len(s)) :

      # Checking character in string
      if (s[i] == c):
         res = res + 1
   return res
for i in letter3:
   Amino.append(i)
   total=0
   for seq in linker:
      cnt=count(seq,i)
      total=total+cnt
   counts.append(total)
for j in counts:
   totsum=sum(counts)
   avgnum=j/totsum
   print(round(avgnum,3))
print(totsum)
```

```
import pandas as pd
df=pd.DataFrame({'Amino Acid': pd.Series(Amino),'Count': pd.Series(counts)})
df.to_excel('AAcount_linkers_wholeprtn.xlsx', index=False)
```

Code to calculate amino acid propensity:

```
import xlrd
import xlsxwriter
import re
from itertools import takewhile
from string import printable
import os
os.chdir(r"C:\Users\svpan\Downloads\thesis res" )
wb2 = xlrd.open_workbook("job665.xlsx")
sheet = wb2.sheet_by_index(0)

linker=[]
whole=[]
for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    whole.append(row[2].value)
    linker.append(row[5].value)

res=[]
linker.pop(0)
whole.pop(0)
for i in linker:
    linker[linker.index(i)]= float(i)
for k in whole:
    index1=whole.index(k)
    whole[whole.index(k)]= float(k)
    res.append(linker[index1]/k)
for j in res:
    print(round(j,2))
```

Code to calculate length and standard deviation of linker lengths:

```
import xlrd
import xlsxwriter
import re
from itertools import takewhile
from string import printable
```

```
from itertools import islice
import math

wb2 = xlrd.open_workbook("red_link.xlsx")
sheet = wb2.sheet_by_index(0)

linkerlen=[]

for i in range(0, sheet.nrows):
    row = sheet.row_slice(i)
    linkerlen.append(row[2].value)
res=[]
cnt=0
linkerlen.pop(0)
j=0
z=30
avg=[]
for i in linkerlen:
    linkerlen[linkerlen.index(i)]= int(float(i))
i=0
stddev=[]
smple=[]
while i<len(linkerlen):
    newlink=linkerlen[j:z]
    tot=sum(newlink)
    nom=len(newlink)
    avg2=tot/nom
    avg.append(tot/nom)
    for i in newlink:
        sd=0
        sd=sd+(i-avg2)**2
    no=1/(len(newlink)-1)
    std=no*sd
    stddev.append(round(math.sqrt(std),2))
    smple.append(str(j)+"-"+str(z))
    i+=1
    j=z
    z+=30
    if z == 7500:
```

```
        break


stavg=sum(stddev)/len(stddev)
tot=sum(avg)
nom=len(avg)
totavg=tot/nom
print(stavg)
print(totavg)
sd2=0
tt=sum(linkerlen)
popavg=tt/len(linkerlen)
for i in linkerlen:
    sd2=sd2+(i-popavg)**2
num=1/(len(linkerlen)-1)
popstd=sd2*num
print(popavg)
print(math.sqrt(popstd))

import pandas as pd
df=pd.DataFrame({'Sample set': pd.Series(smple),'Sample mean': pd.Series(avg),'Sample sd':
pd.Series(stddev)})

df.to_excel('Linker length analysis.xlsx', index=False)
```