AUTOMATED TEXT MINING AND RANKED LIST ALGORITHMS FOR DRUG DISCOVERY IN ACUTE MYELOID LEUKEMIA

AUTOMATED TEXT MINING AND RANKED LIST ALGORITHMS FOR DRUG DISCOVERY IN ACUTE MYELOID LEUKEMIA

By DAMIAN V. TRAN, B.Sc.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements

for the Degree of Master of Science

MASTER OF SCIENCE (2019) McMaster University

Faculty of Health Sciences – Biochemistry Hamilton, Ontario

Title:	Automated Text Mining and Ranked List Algorithms for Drug Discovery in Acute Myeloid Leukemia
Author:	Damian V Tran BSc
Supervisor:	Dr. Kristin Hope PhD
Number of pages:	xvi, 141

Lay Abstract

Lead generation is an integral requirement of any research organization in all fields and is typically a time-consuming and therefore expensive task. This is due to the requirement of human intuition to be applied iteratively over a large body of evidence. In this thesis, a new technology called the Artificially-intelligent Desktop Assistant (AiDA) is explored in order to provide a large number of leads from accumulated biomedical information. AiDA was created using a combination of classical statistics, deep learning methods, and modern graphical interface engineering. It aims to simplify the interface between the researcher and an assortment of bioinformatics tasks by organically interpreting written text messages and responding with the appropriate task. AiDA was able to identify several potential targets for new pharmaceuticals in acute myeloid leukemia (AML), a cancer of the blood, by reading whole-genome data. It then discovered appropriate therapeutics by automatically scanning through the accumulated body of biomedical research papers. Analysis of the discovered drug targets shows that together, they are involved in key biological processes that are known by the scientific community to be involved in leukemia and other cancers.

Abstract

Evidence-based software engineering (EBSE) solutions for drug discovery that are effective, affordable, and accessible all-in-one are lacking. This thesis chronicles the progression and accomplishments of the AiDA (Artificially-intelligent Desktop Assistant) functional artificial intelligence (AI) project for the purposes of drug discovery in the challenging acute myeloid leukemia context (AML). AiDA is a highly automated combined natural language processing (NLP) and spreadsheet feature extraction solution that harbours potential to disrupt the state of current research investigation methods using big data and aggregated literature. The completed work includes a text-to-function (T2F) NLP method for automated text interpretation, a ranked-list algorithm for multi-dataset analysis, and a custom multi-purpose neural network engine presented to the user using an open-source graphics engine. Validation of the deep learning engine using MNIST and CIFAR machine learning benchmark datasets showed performance comparable to state-of-the-art libraries using similar architectures. An n-dimensional word embedding method for the handling of unstructured natural language data was devised to feed convolutional neural network (CNN) models that over 25 random permutations correctly predicted functional responses to up to 86.64% of over 300 validation transcripts. The same CNN NLP infrastructure was then used to automate biomedical context recognition in >20000 literature abstracts with up to 95.7% test accuracy over several permutations. The AiDA platform was used to compile a bidirectional ranked list of potential gene targets for pharmaceuticals by extracting features from leukemia microarray data, followed by mining of the PubMed biomedical citation database to extract recyclable

pharmaceutical candidates. Downstream analysis of the candidate therapeutic targets revealed enrichments in AML- and leukemic stem cell (LSC)-related pathways. The applicability of the AiDA algorithms in whole and part to the larger biomedical research field is explored.

Acknowledgements

As Isaac Newton had long ago so eloquently expressed, we all stand on the shoulders of giants while we speak proudly about our accomplishments. The work that I put forward in this thesis is no exception to this fact and was only possible due to the support and flexibility of the giants that have lifted me to the place I am today.

I would like to offer my gratitude to Dr. Kristin Hope, my supervisor since the summer of my 3rd undergraduate year at the McMaster Stem Cell and Cancer Research Institute. It was her thoughtful, intelligent advice and her willingness to embrace crazy, whacky ideas that allowed me to combine my love for science with my mind's ridiculousness to make AiDA happen. She's seen me grow through my various academic phases, from my beginnings in the wetlab to where I am now, farther than ever from a pipette. I feel privileged to have had Dr. Hope as a supervisor—if the dice had landed another way and I had worked for someone else it's likely none of these technologies would even exist today.

I would also like to thank Dr. Andrew McArthur, one of the members of my graduate committee, for his support and belief in my project. I had once walked into his office with nothing but a crazy idea about a neural entropy model and he was willing to lay down time and (discuss) expensive server infrastructure to make it happen—if that's not faith I don't know what is. Last not least, I'd like to thank Dr. Brian Leber for committing his time to offer his advice on this project as the third member of my committee. As busy a man as he is, I feel privileged to have been able to hear the input

of a medical practitioner whose working mission is to save and extend human lives on the daily.

Looking back in time, I'd like to thank my former graduate student supervisor Dr. Laura DeRooij, who had overseen my work during my undergraduate thesis and had offered continual support and advice in the early days of my research career. Aside from her research brilliance, another characteristic I had only post-emptively grown to fully appreciate was her patience. I had once misinterpreted a label on her primers which led me to believe that *ELAVL1* contained mysterious characteristics with an *H-type* nomenclature, which I never could elucidate no matter how much I dug into the literature. I only found out a year later that this was in fact a number sign… and yet we had been continually adopting this [H1, H2, H_n, …] labelling schema the entire time on everything from our PCR spreadsheets to our cell cultures. It was then that I learned two lessons: 1) the value of asking questions no matter how dumb they may sound, and 2) how go-with-the flow Laura truly was. My work to this day has been heavily influenced by her efforts to develop a holistic functional genomic screening method for cancer. It gave me ambitious visions for a bigger scale of study solely from what was once, like AiDA, just an insanely whacky idea.

On that note I'd like to thank the rest of the Hope Lab, the members of the SCCRI, and the many researchers that I had consulted with throughout my condensed journey of learning over the last couple years. AiDA was, *fittingly,* an iterative process of reinforcement learning where I learned to discover a set of needs that prevailed in the research community. As I consulted with others, obtained help and valuable feedback,

and helped them with their projects, I learned where to divert the most attention during the development process. In a way this community of researchers had been my first "customer validations" regarding the need for a simpler interface to bioinformatics.

I owe much gratitude to my family for my successes: my father who had escaped war and tyranny in Vietnam, risking his life at sea for a better life for himself, and eventually myself and my sister. My mother, who left humble beginnings in rural Russia to begin a new life in Canada, eventually finding her way into teaching administration. Both arrived at some point barely knowing the language, and yet they etched out a living that supported my sister and I to realize the dreams that they once had as children. I have endless respect for them, and the many other immigrants who risk everything to try something new here in this beautiful country.

Finally, I'd like to give my greatest appreciation to my love, Hala. She's been with me since the very beginning of this journey and has been endlessly supportive even during my long days and nights of non-stop development. There were many points where my path had been uncertain, where it was unknown if all of this could even lead to a sustainable life medium. Before the technology, the scholarships, the contracts, and the business, I was just a kid who talked too much about science but with little to his name (quite literally, I had struggled even to afford a plain bagel and a coffee at the time). I had been an undergrad struggling to make ends meet, working 30 hours a week as an underpaid café barista to afford living and commuting while finishing my education.

As busy (and as broke) as I was, she stuck by my side with conviction that, despite the obstacles that had been laid in our way, we would make it through. Never—not once—has she failed to wholeheartedly believe in me even during my multiple pivots in career decision-making that lead me to my current place. We're far from the clear, and certainly many obstacles of unimaginable scale and complexity lie ahead as we grow together and I push to bring this technology to the world, but there's no other person I'd rather have by my side than her. She's a breath of fresh air, the spice of life, and reminds me that there's more to existence than logic gates, activation functions, and the Smith-Waterman algorithm—and that's saying a lot because I love the Smith-Waterman algorithm more than a lot of things.

Lastly, thank you, the reader, whomever you may be—if not one of the aforementioned persons—for taking the time to read and learn something new. I spent the last two years of my life hitting my head against a wall that I had no idea could even break. I put together pieces that I had no idea would even fit. In my mind this computational thesis published in requirement of a health sciences degree captures the essence of what science has always been: a diversion from the norm of everyday life to try new things in the effort of making a change for the better. As the "norm" in the polynomial surface of the function of society is a deep, heavy-set local minimum, we require disruptive force to push it out of the status quo and into an even lower energy state where life has been completely and unrecognizably transformed. I am excited to participate in the promising movement that is artificial intelligence, and I hope you'll find something of value from

among this chronicle to make your own contributions to our growing, rapidly developing world for the better.

Table of Contents

List of Figures, Tables, and Appendices

Abbreviations and Symbols

AI	Artificial intelligence
AiDA	Artificially-intelligent Desktop Assistant
ANN	Artificial neural network
AML	Acute myeloid leukemia
API	Application programming interface
CMP	Common myeloid progenitor
CNN	Convolutional neural network
DOM	Document object model (HTML)
EBSE	Evidence-based software engineering
ES	Enrichment score (GSEA)
FAB	French-American-British (AML classification)
FACS	Fluorescence-activated cell sorting
FDR	False discovery rate
FWER	Family-wise error rate
GDB	GNU project debugger
GEO	Gene expression omnibus (NCBI)
GMP	Granulocyte-macrophage progenitor
GNU	GNU's Not Unix (recursive)
GSEA	Gene set enrichment analysis
GUI	Graphical user interface
HTML	Hypertext markup language
HSC	Hematopoietic stem cell
HUGO	Human Genome Organization (South Korea)
I/O	Input/output
ITD	Internal tandem duplication
LE	Leading edge (GSEA)
LSC	Leukemic stem cell
MEP	Macrophage-erythroid progenitor
MLP	Multilayer perceptron
MPP	Multipotent progenitor
MSigDB	Molecular Signatures Database (Broad Institute, USA)
NCBI	National Center for Biotechnology Innovation (USA)
NES	Normalized enrichment score (GSEA)
NGS	Next-generation sequencing
NLP	Natural language processing
PAM	Prediction analysis of microarrays
PDF	Probability density function
PMF	Probability mass function

PMID	Pubmed ID
RAM	Random access memory
RES	Running enrichment score (GSEA)
RPKM	Reads per kilobase-million
RSEM	RNAseq by expectation maximization
SGD(M)	Stochastic gradient descent (with momentum)
SLR	Systematic literature review
T2F	Text-to-function
TCGA	The Cancer Genome Atlas (NIH, USA)
TTD	Therapeutic Target Database (BIDD, Singapore)
XML	Extensible markup language

Declaration of Academic Achievement

I, Damian V. Tran, declare this thesis to be my own work, and am the sole author of this document. No part of this work has been published or submitted for publication or for a higher degree at another institution.

To the best of my knowledge, the content of this document does not infringe on anyone's copyright. All copyrighted figures included in chapter 1 are the property of their respective owners, cited directly below where they appear, where applicable.

My supervisor, Dr. Kristin Hope, and the members of my supervisory committee, Dr. Andrew McArthur and Dr. Brian Leber, have provided guidance and support during many stages of this project. All development of original algorithms, modification of existing algorithms, and algorithm validation was performed by me.

Chapter 1: Introduction

1.0: Preamble

The core directive of this thesis was to demonstrate that next-generation sequencing data extraction and deep natural language processing could be applied to create an end-to-end lead generation platform. It is a work that combines several components of bioinformatics, stem cell science, clinical oncology, and deep learning in an unlikely combination to produce a highly automated, targeted solution for research investigation. Regression statistics and deep learning are sequentially applied in the endeavor of automating specific niches of human intuition such as functional responses to text and contextual recognition. The deep learning technologies applied in this work, discussed in Chapter 2, were implemented using a custom neural network engine created in the C++ programming language. Discoveries made by this combination of new technologies will be covered in detail in Chapter 3, which include validations of the custom deep learning engine, the regression methods, and the predictions made by the system. The core translatability of the concepts discussed in this work will be discussed in Chapter 4, most of which revolve around integration with the desktop application called the Artificially-intelligent Desktop Assistant (AiDA). The AiDA platform, and many of the algorithms integrated as part of this work, were built using the CVision and HyperC open source libraries, which are contributions to the development community provided free of any charge (or requirement for any attribution).

In this chapter, several required concepts will be covered at the surface level to acquaint the reader with the knowledge required to understand the methods descripted in Chapter 2, and the results reported in Chapter 3. Supplementary visuals will be provided alongside the text in pertinent areas—in cases where the images have not been originally created for the purposes of this thesis, the original authors have been cited. It is highly recommended that the reader follows along with citations in this chapter in areas where they are not familiar with the content, in order to strengthen their understanding of the core concepts. As a high-level summary, the concepts of this chapter include:

- Natural language processing for biomedical research
- The relevance of next-generation sequencing analytics to lead generation
- Chatbots in the enhancement of the user experience (UX)
- A prelude to deep learning theory using neural networks
- Background on the Acute Myeloid Leukemia cancer, and the relevance of stem cells to its severity

This chapter begins with an introduction to natural language processing, and the logic behind the automation of data extraction from unstructured text data.

1.1: Automatic text mining in bioinformatics

An essential task in all fields of research is the identification of viable leads either directly or indirectly supported by accumulated evidence for the purposes of investigating novel avenues of study and/or justifying the continuation of current research projects. The most thorough method of accomplishing this today is the systematic review process,

which involves an aggregation of evidence from the scientific literature through systematic literature review (SLR) and experimental data through data analytics. Systematic review currently incurs a high time cost, and therefore financial cost, due to the inefficiency and laborious nature of both SLR and data analytics (Higgins and Green, 2008; Chapman *et al.*, 2010; Jonnalagadda *et al.*, 2015). The process of hypothesis generation, often accomplished by a combination of systematic review, previous work, and open source data analysis, is a major determinant for the curation of funding for future research. For many research groups, long-term success is mediated by consistency in identifying and following research leads, among other critical factors such as the competency of the team as a unit and effective management of time and resources. On a societal level, consistent, quality hypothesis generation fuels a chain of innovation that over time drives national economic growth through the evolution of ideas into basic research, which are then followed by translatable proof-of-concepts and eventually productive, profitable solutions.

The Cochrane guideline for extensive SLR suggests that a single investigator invests a mean total review time of 6-8 months, with an upper limit of one full year (Higgins and Green, 2008). Less extensive exploratory literature reviews poised at satisfying simpler questions take an average of 26.9 hours (Bullers *et al.*, 2018). Time measurements of literature review tasks in a randomized sample of librarians indicated that the search, interpretation, and writing components of formal systematic reviews explained much of the variability in the time taken to completion (Bullers *et al.*, 2018). The need for automated software solutions for SLR has been formally recognized since 2004, spurring

forward the field of Evidence-Based Software Engineering (EBSE) (Dyba *et al.*, 2005). A number of freeware and commercial software tools have been created to attempt to mitigate the time costs of SLR, such as the EPPI-Reviewer (Thomas and Brunton, 2007), SLR-Tool (Fernández-Sáez *et al.*, 2010), TrialStat SRS (trialstat.com), and most recently DistillerSR (Evidence Partners, 2011). All of these solutions are closed-source software that do not offer internal API tools for redistribution. Those that are commercial come with substantial price tags that do not reduce the combined time-financial costs of SLR. Others have recognized that accessible solutions in EBSE have not been created, particularly those that can track literature searches and analyze clinical data (Brogger, 2007).

To help scale the use of NLP in large-scale literature searches, software engineers have created syntactical search tools such as Agilent Literature Search (https://www.agilent.com/labs/research/litsearch.html), which make use of NLP to help the user find information related to a number of criteria. Cytoscape is the most well-known way to visualize gene interaction information, and has integrated Agilent Literature Search into its app engine to create gene interaction networks bolstered by NLP (Shannon *et al.*, 2003). As powerful as the duo may be for early hypothesis generation, the extensive syntax of the Agilent literature search tool and the method to merge its outputs with the Cytoscape front-end present a significant learning curve to the end user. Considering that the intended end user may have limited background in the syntax of computer logic, significant time investment is required for users to bring themselves up to speed enough to install the tool, peruse the documentation and learn the

syntax, and then deploy it. As a result of these roadblocks to user engagement, Agilent's literature search been relatively underused by the biomedical research community compared to other packaged solutions such as the Broad Institute's GSEA tool (Subramanian *et al.*, 2005).

In addition to small-scale lead generation, a need has been acknowledged for high-throughput large-scale text mining solutions in the bioinformatics community (Ivanisenko *et al.*, 2015; Labaer, 2003; Spangler *et al.*, 2014). The challenges faced in small scale hypothesis-generation are exponentially exacerbated when the same processes are applied on large, automated runs across the accumulated literature. At the time of writing there are over 29 million citations indexed by the PubMed citation database, a number which demands that any holistic analysis of the accumulated biomedical evidence body be fully automated. Full automation of the literature experiences many seemingly insurmountable obstacles due to the variability of unstructured text formats in the literature. Publication formats differ between journals, as well as within journals (ie. letters to the editor, communiqués, review papers, methods papers, and results papers). An effective EBSE solution for large-scale automated text mining requires that these diverse text input types be standardized in order to produce standardized inferences and responses. Currently, the state-of-the-art is a combination of manual curation and automated lead generation which involves the extraction of key words from a large body of papers and the ordering of papers according to key word scores (Delen and Crossland, 2008; Franceschini *et al.*, 2012; Ivanisenko *et al.*, 2015). Databases such as STRING incorporate automated text mining to help fill the gaps in manually curated data, thus

speeding up the process while introducing an acceptable margin of automation error (Franceschini *et al.*, 2012). These methods proved to be invaluable in speeding up the process of biological pathway reconstruction but lacked the ability to discern the directionality and type of interactions between named biomedical entities. The most recent technology to address automated text mining needs is ANDSystem, which implements a dictionary-based parsing model to identify named entities in the literature and integrate new discoveries with accumulated pathway knowledge (Ivanisenko *et al.*, 2015).

All present automated literature-mining methods require an abundance of present knowledge to help fact-check discoveries made by the forward algorithms. This is because the ability to discern the main idea and context of a body of dense literature is, at the time being, a purely human ability. There are many variables with many degrees of freedom that require deep inference to identify. When considering that the only free information available about the vast majority of publications emerges from their title and abstracts alone, computational algorithms become thrown against challenges that even most humans have difficulty discerning. When perusing the literature, for each publication a reader must be able to correctly identify:

1. The main idea of the work
2. The contextual background of the work
3. The discoveries made as a result of the work
4. The validity and confidence of the work's results

We can partially satisfy requirements (1) and (3) using the methods cited previously but struggle greatly with requirements (2) and (4). Consider the search for the gene locus coding for the pyruvate carboxylase enzyme, abbreviated "PC." There is an enormous diversity of abstracts cited in the PubMed database abbreviating other word combinations for "PC," such as prostate cancer (Kamisawa *et al.*, 2008), pancreatic cancer (Fattahi *et al.*, 2009; Horvath *et al.*, 2001), and phosphatidylcholine (Amtmann, 1996; Exton, 1990). One might suggest the intuitive solution of searching purely for the name of the gene product, however many abstracts do not define the names of their abbreviations and thus this kind of search would prove to be too conservative. If not otherwise defined, the way a human would be able to discern between the proper definitions of an acronym across a range of different papers would be through the context of the text. A paper failing to define an acronym in a general study of democratic decline across the globe would certainly signify that the acronym's definition is unlikely to correspond to a biochemical definition, and thus an educated inference can be made about an alternate meaning based on that probability. An automated solution that can perform that kind of probabilistic inference would allow for a vast improvement in the coverage of today's manually curated biomedical databases.

1.2: Big data analytics in hypothesis generation

Robust, reproducible data analytics are, alongside SLR, an integral component of evidence-based systematic review. While the literature serves the purpose of validating findings based on collective reports and opinions founded upon experimental data, it is impossible to move forward along new lines of investigation without some kind of new

input in the form of novel experimental evidence, or reanalysis of existing evidence in innovative ways. Therefore, it would be a basic requirement of a true hypothesis-generating AI that it be able to peruse the body of experimental evidence in addition to the accumulated literature.

Several tools have been published to help users manually extract and visualize information from raw experimental data. These include the R programming language (Ihaka and Gentleman, 1996) and associated microarray analytics packages such as Limma (Smyth, 2005). Web developers have also assembled public web tools such as the cBIOPortal for cancer genomics (Gao *et al.*, 2013) and the BloodSpot gene expression platform for hematopoiesis (Bagger *et al.*, 2015). Other tools for higher-level population analytics exist such as the Pathway Commons centralized biomedical pathway hub (Cerami *et al.*, 2010), the Gene Ontology unified biology database (Ashburner *et al.*, 2000), and the DrugBank drug discovery web tool (Wishart *et al.*, 2006).

Since these tools are scattered between multiple sources on multiple platforms, it requires much background knowledge, some technical expertise, and time spent finding and/or sourcing the tools to piece together a complete virtual pipeline that can adequately analyze experimental data for a single potential research lead. There are no tools that chain together all these software options into one solution, let alone one that can allow for sequential analyses of multiple leads. Likely barriers to the creation of a unified solution would be the complexity of consolidating information across multiple platforms, and potential cluttering that would arise from the required user interface. A platform unifying

all these tools would turn multiple learning curves into a single learning curve associated with learning how to use the aggregated interface.

A major challenge to consolidating these kinds of tools however is the maintenance of simplicity in the user experience (UX). UX design has more recently become a major priority for developers in many industries, especially as the demands for versatility in each software product increase over time (Gray *et al.*, 2015; Øvad and Larsen, 2015; Unger and Chandler, 2012). While relatively simple interfaces such as literature searches can be presented in the form of a search bar with "advanced options," the degrees of freedom required in deep data analytics would make such a page expand to unreasonable depth. For example, when the visualization of categorical data in a spreadsheet is required, the typical path of extraction would follow:

1. If data is not normalized and/or formatted, push the dataset through a pre-processing toolchain
2. Find the labels associated with each category in the spreadsheet
3. (If annotations are not present) look for annotations in another file included with the data package
4. Determine the organization of the annotations (headers vs. body)
5. Determine the orientation of the annotations (row vs. column)
6. If required, match annotations to IDs in the raw data spreadsheet.
7. Seek along the required rows/columns for the location of the desired data values
8. Perform a dictionary lookup if the data points have been hashed or tagged
9. Extract the data values into vector format for each category

10. Perform logistic/linear regression statistics on the selection against the background
11. Visualize the data, supply supplementary statistics for reporting

The enormous amount of customizability that follows along each step makes it nearly impossible to create a tool with an options panel that is both intuitive and flexible enough to capture the range of user demands. Each of these steps may vary to a greater extent if the data is presented in a non-standard format, the data labels and data points are separated into different files, or the data points are labelled using a hash or conversion table (for example, microarray data). Therefore an intuitive user experience would require that the application be able to foresee the users' needs based on the many baseline criteria of the dataset and experimental question.

1.3: Chatbots in modern applications

A supplement—and sometimes alternative—to classic graphical user interfaces is the emergent chatbot technology powered by deep learning. Chatbots have the potential to simplify the user experience by removing rarely-used buttons and menus from the interface, and providing a more fluid interface to functions that are difficult to control with conventional GUI methods (Dove *et al.*, 2017). Chatbot technology as a deployable asset is still in its infancy and being actively experimented with by many corporate and research organizations to find the best-fitting use cases. The more modern push for chatbot integration into modern workflows arises around market reports that proper integration of chatbot technology can reduce customer service costs through enhanced accessibility to routine information (Reddy, 2017).

A recent UX poll investigated users' motivations for using online chatbot technology and found that overall, users reported that they perceived an overall increase in productivity, and better access to timely information (Brandtzaeg and Følstad, 2017). Despite the incentives, IBM's Watson recently experienced disappointing numbers in its AI-related sales revenue last year (Green, 2018). Limitations of IBM's Watson solution include lack of a localized hardware option, inaccessibility to individuals and small businesses, and difficulty in learning and integration (CompareCamp, 2019; Jarvis, 2019). Other competing solutions such as PandoraBots' Mitsuku (pandorabots.com/mitsuku) and Rollo Carpenter's Cleverbot (cleverbot.com) are technically impressive but have not gained traction as portable solutions because they have not yet been demonstrated in functional contexts. More functional solutions such as Siri (Apple Computers, Inc.), Alexa (Amazon, Inc.) and the Google Assistant, have seen success among larger client bases but do not have lower-level APIs accessible to developers to perform more specific tasks (with the exception of Alexa).

State-of-the-art chatbot technologies are created using deep learning methods involving the conversion of words into a variable-dimensional (n-dimensional) vector called a word embedding. The problem space in natural language processing is immense due to the sheer number of word syllables, semantic ordering, and potential for mis-spelt words and non-conventional grammar. By classifying words based on many orthogonal "features" during the process of "feature extraction" it's possible to reduce the number of degrees of freedom, and therefore the training time of learning models.

The model fitting process in NLP includes the necessary splitting of information into test and training datasets, which themselves consist of sentences mapped to a desired output. If this system is set up to map sentences to required actions, it's possible to simultaneously guarantee that a large array of possible requests will be understood by the chatbot while forecasting response accuracy to unseen user requests. Furthermore, the deep learning NLP method is not only a powerful, development technique but also a robust, measurable automated testing protocol.

1.4: Deep learning using artificial neural networks

Artificial neural networks (ANN) are computational mimics of the biological method of signal transduction (ie. real neurons). The idea behind non-linear signal transduction was first conceived by Warren McCulloch and Walter Pitts in 1943 (McCulloch and Pitts, 1943). They postulated that more complicated, non-linear logic could arise from math emulating the neural "all-or-none" action potential system. This train of thinking wasn't well applied at the time since the means to cognate such complex systems was as of then far out of reach. Donald Hebb iterated upon this thinking when he made the original publication of his book: *The Organization of Behaviour: a Neuropsychological Theory,* in 1949 (Hebb, 1962). In doing so, he established a method of unsupervised learning called "Hebbian learning." He postulated that a field of virtual neurons could be corrected into a specific desirable configuration by activity-dependent synaptic modification. Research had begun on emulating these models in early digital computers, and in 1954 the first self-organizing computational system was described (Farley and

Clark, 1954). The activation parameters for this kind of network were initially defined as:

$$h_j(t) = h_{max}\exp(-a_j t) + h_{jmin} + h_{bias}(t) \quad (9)$$

$$\Delta s_j(t) = -b_j\, s_j(t-1) + \sum_i w_{ij} \quad (9)$$

(Farley and Clark, 1954)

Where $h_j(t)$ represents the threshold function and $\Delta s_j(t)$ represents the change in excitation for a node at index j at time t. As this network was only a single layer deep, a single bias (h_{bias}) was applied to offset the distribution of the input signals into a more favourable range for the activation function. The concept of "signal decay" had been considered (represented by b_j), which was a method of avoiding divergence in the system.

Multi-layer models were proposed by Russian data scientists Oleksiy Ivaknenko and Grigor'evich Lapa, which Ivaknenko had integrated into his "group method of data handling" (GMDH) (Ivakhnenko and Lapa, 1967). GMDH is a computational inductive model that tries to minimize the output of a complicated base function derived from a multilayer system. Its main purpose is to fit non-physical models to multi-parametric datasets and allows for the completely automatic tuning of many trainable parameters (A.G. Ivaknenko and G.A. Ivaknenko, 1995). The base function is split into smaller partial models with coefficients that are estimated by least-squares regression. This method is still used today and has even proved to be more effective than some contemporary neural methods at solving time-series forecasting problems (Li *et al.*, 2017). The most popular of the base functions used in the GMDH method is the

Kolmogorov-Gabor polynomial support function, which is a high-degree polynomial based on the Volterra series with many variable contributors:

$$y = a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{M}\sum_{j=1}^{M} a_{ij} x_i x_j + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} a_{ijk} x_i x_j x_k$$

(A.G. Ivaknenko and G.A. Ivaknenko, 1995)

In this function, the vector of inputs X(x_i, x_j, x_k, …) is split into layers (i, j, k, …) that are modulated by a vector of input weights A(a_i, a_j, a_k, …) and a single bias a_o to produce output y. This system is corrected by a genetic-like system of permutating a set of candidate models and selecting those that perform the best.

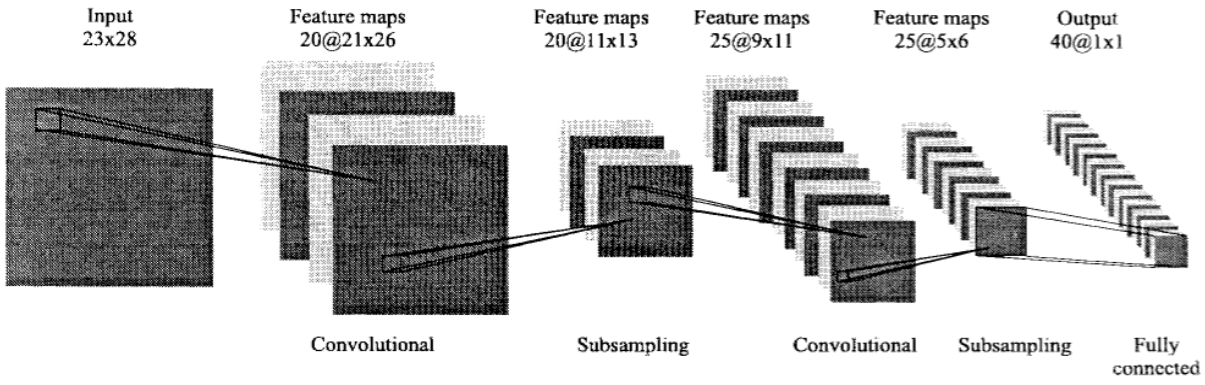
There were several issues with these neural optimization methods, mostly centered around the large number of trainable parameters, that had made it infeasible to produce and deploy them with the available computational hardware. Deep learning research had thus experienced a hiatus until 1974 when Paul John Werbos, a doctoral student in the social sciences, devised a clever solution to multi-parametric optimization called backpropagation (Werbos, 1974). His PhD dissertation had delineated a mathematical loophole to multi-layer error correction that involved the sending of a reverse signal from the outputs to the input layer carrying differential magnitudes of error blame. Instead of attempting to calculate the derivative of a complex polynomial system in order to optimize a model with hundreds, potentially thousands of parameters, Werbos postulated that each parameter could be assigned a portion of the output error during the backward progression of this signal that would then be carried over to other dependent parameters.

By splitting the signal, many smaller, less computationally expensive derivatives would be computed that would be applied at the end of the training batch to increment the model in the direction of lowest error. The signal would be modified after passing through each parameter proportionally to the derivative of the forward signal. Werbos' dissertation had never been published but contained work that laid the foundation for countless iterations of modern deep learning publications. It remains one of the most highly cited unpublished works to this date (4999 citations as of July 2019).

Following Werbos' discovery of the backpropagation algorithm was a vastly accelerated push into neural network research using the exponentially-advancing power of micro-transistor computing. The idea of distributing the processing of error was thus named "parallel distributed processing," which was first reported in David Rumelhart and James McClelland's work using the fully connected multi-layer neural network, or multilayer perceptron (MLP) (Rumelhart, 1986). This movement in cognitive computing was also called "connectionism", due to the adherence to full connectivity among the layers of artificial neurons of fully connected networks. These MLP networks found utility in modelling tasks of high complexity such as the prediction of protein secondary structures. Over the course of 7 years, bioinformaticians had progressed from the conception of the system to the prediction of transmembrane helices from primary structure information with over 95% accuracy (Qian and Sejnowski, 1988; Rost and Sander, 1993; Rost *et al.*, 1995).

Complex image processing tasks were vastly simplified when powered by a concept called "max-pooling," which was first applied to the automatic segmentation of 3D

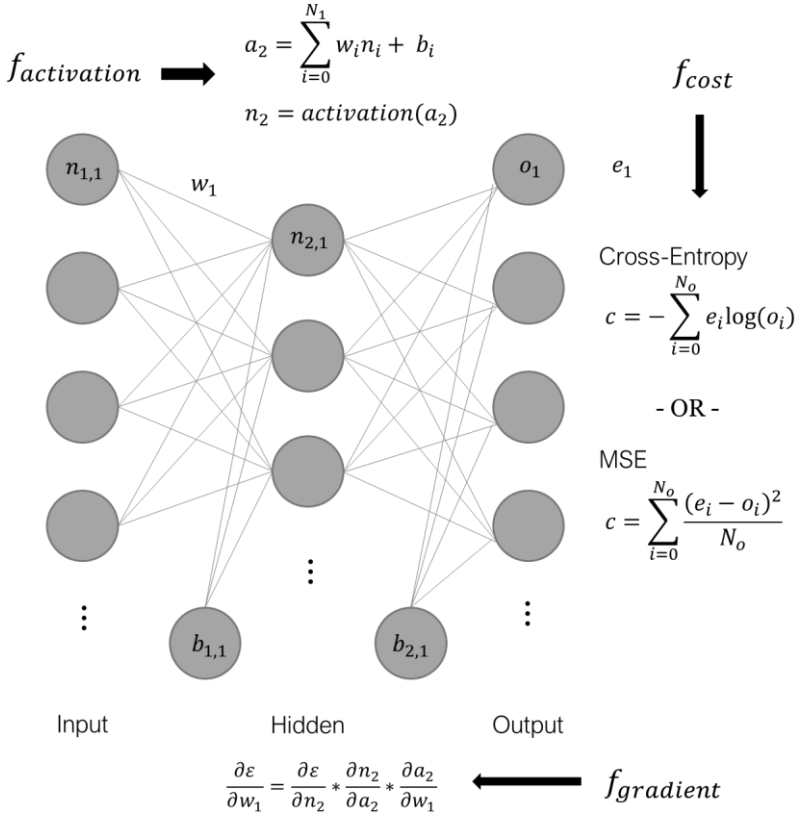
images (Weng *et al.*, 1992, 1993). The pooling method sub-samples an input of dimensions (n_x,n_y) into one of (n_x/p_x, n_y/p_y) where p represents the pool size. Within each pool, the pixel with the highest intensity represents the pooled pixel, and all other information from the pool is discarded. This introduces "shift invariance" to the internal construct of a neural model, allowing it to make the same predictions for input data that has been transformed within the variability of the pooling range (typically, a 2x2 or 3x3 coordinate range). In 1997, to satisfy industry demands for reliable identity verification biometrics, Lawrence at al. developed the "convolutional neural network" (CNN) concept that applies a series of image transformations to extract pertinent patterns from faces before feeding it into an MLP network (Lawrence *et al.*, 1997). The CNN was a diversion from classic connectionism which introduced pre-processing by layers of neurons that were not fully connected, but instead shared a kernel of weights together. This pre-processing was necessary to capture the enormous variability in face images that arises from direction, lighting, expressions, hair, and color. The authors of the CNN work had surmised that images could be simplified by extracting key features from the images while throwing out the remainder of confounding noise.



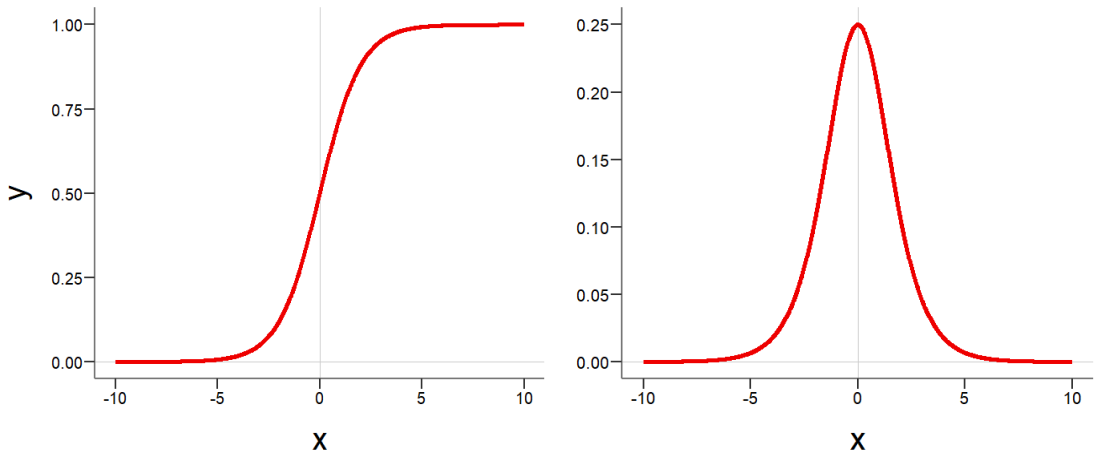
(Lawrence *et al.*, 1997)

The diagram above demonstrates how this is performed, written in 1997 but still applied in the same way today for tasks such as digit recognition on bank notes and cheques (Holi and Jain, 2019; Pham *et al.*, 2017; Srivastava *et al.*, 2019). Convolution of an input image of size (n_x,n_y) becomes altered and compressed by a convolutional filter of size (k_x,k_y) into a slightly smaller feature map of size $(n_x – k_x – 1, n_y – k_y – 1)$. Max-pooling of the feature map by dimensions (p_x,p_y) then provides an even more condensed feature map of $((n_x – k_x – 1)/p_x, (n_y – k_y – 1)/p_y)$. Through successive processing and compression cycles a relatively large input image containing 644 unique pixels is simplified by multiple cycles of convolution and pooling such that it can be read by a layer of only 40 neurons. Previous methods would have required 644 input neurons to read the image in order to feed it to the deeply connected MLP. Thus the CNN system not only adds additional translational invariance and feature sensitivity, but also greatly simplifies the fully connected component of deep learning models.

Modern neural networks have become extremely diverse in their implementations, with a variety of different activation functions, cost functions, architectures, and optimizers. The canonical MLP architecture can be diagrammed as follows:



The key mathematical mechanisms in an MLP are the activation function ($f_{activation}$), the cost function (f_{cost}), and the gradient ($f_{gradient}$). The activation function is a two-step function that consists of the summation of incoming weighted signals and bias (a_1) followed by a differentiable non-linear function such as the sigmoid transformation (Cybenko, 1989). The sigmoid function was one of the earliest non-linearities to be introduced into cognitive computing, proposed due to its regions of first order sensitivity at the input extremes.



Shown above are the graphs of the sigmoid function (left) and its derivative (right). The sigmoid function with no transformations ranges from $0 < y < 1, y \in \mathbb{R}$, and as a result "squashes" inputs of all ranges into a standardized range that can be learned by deeper layers in the network.

Once the output layer of the MLP has been activated, the value of the cost function (often referred to as the "loss" function) can be calculated. The cost function is a metric of how erroneous the current network inference is and is proportional to the difference between the expected and observed values at each output neuron. The two most popular cost functions are the mean squared error of outputs (Scalero and Tepedelenlioglu, 1992; Specht, 1991) and more recently the cross-entropy loss function meant to be used together with the "Softmax" activation function (Hautamäki *et al.*, 2013; Kline and Berardi, 2005; Zhang and Sabuncu, 2018). The derivative of the loss function is the substrate for the initialization of the backpropagation signal, and is calculated for each output neuron and sent back in a step-wise manner through the entirety of the network.

The backpropagation phase involves the differential, step-wise assignment of "blame" to every trainable parameter in the network (weights, biases, batch normalization shifts, and others). The rate of change of error with respect to each output neuron is the seed for the signal which accumulates at each node in the network, allowing for computationally-efficient alteration of each parameter via the "delta rule."

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \eta\delta(n)\mathbf{i}^T(n) \qquad (1)$$

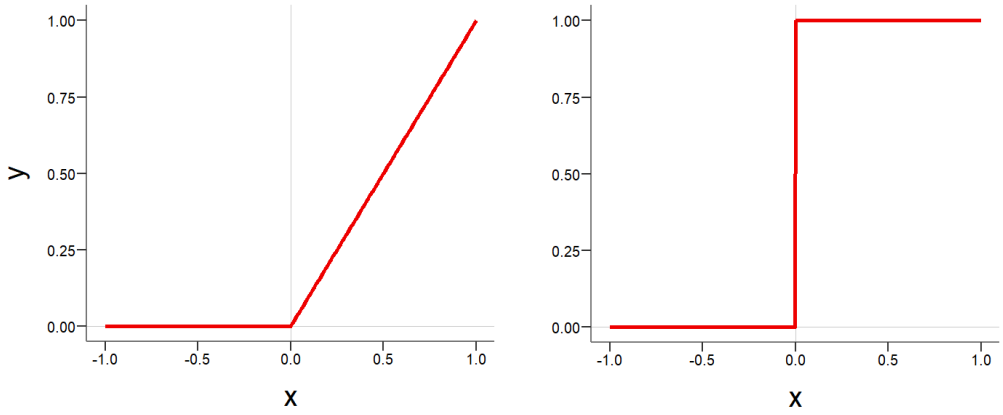
(Stone, 1986)

Exemplified above in Greg Stone's overview of the delta rule for the earliest parallel distributed processing models. The value of each trainable parameter at trial n ($W(n)$) is incremented by the derivative of the cost function at trial n ($\delta(n)$) multiplied by the input $i^T(n)$, collectively the *delta*. The derivative, for both the mean squared error and cross-entropy loss functions, simplifies to the difference between expected and observed values. The delta is modulated by the learn rate hyperparameter η, which tunes the distance of each step taken during the backpropagation process. As the delta rule is only a close approximation of the total system derivative, a learn rate of 1.0 would result in undesirable non-convergence, and even divergence, of the system. Typically, the learn rate is tuned to orders of magnitude between 10^{-2} and 10^{-4} such that the system follows a gentler error function surface on its search for a global minimum (Huang and Stokes, 2016; Krähenbühl *et al.*, 2015; Shin *et al.*, 2016).

There are several functional, architectural, and hyperparameter decisions that can be made before the training process to attempt to fine-tune the convergence of a neural network system toward the theoretical zero error limit. The first, and most obvious, is the method of activation at each layer of the network. Layers may be activated differently across the network, but all neurons of a layer tend to share the same activation function in order to ensure that the entirety of the neural network behaves in a predictable way. It has been widely acknowledged that some form of non-linearity in the activation is necessary to give networks enough density in the function space to model any range of data complexity (Chen *et al.*, 1990; Glorot *et al.*, 2011; Maas *et al.*, 2013). The simplest explanation for this is that linear functions can only model other linear functions, and as such a network of infinite depth will inevitably collapse into a linear function that is the sum of its parts. By introducing ranges of the first order derivative that have resistance to motion, we provide opportunities for different step sizes, and thus the network may shift amorphously.

This resistance to motion was initially a desirable attribute of the sigmoid function (Chen *et al.*, 1990). The "squashing" of the input signal, while a convenient normalizing metric, results in a phenomenon called "gradient vanishing" due to the near-zero value of the derivative as x diverges to positive and negative extremes (Hochreiter, 1998). Following this, Rectified Linear Units (ReLUs) have become increasingly popular as the non-linear component of the activation function due to their ability to introduce complexity into the network model while being computationally lenient (Dahl *et al.*, 2013; Nair and Hinton, 2010; Zeiler *et al.*, 2013). The implementation of the ReLU is straightforward, as below:

$$n_2 = \max(a_1, 0)$$



Shown above are the graphs of the ReLU function (left) and its derivative (right). The nonlinearity is imposed by the restriction of all actionable values of y to positive values. This type of activation more closely resembles the biological "all-or-none" system mediated by neural activation gates in the axon hillock. The shortfall of this method is that, due to the high likelihood of zero gradient deltas, the network may experience "neuron death" over long training runs as neurons run into long regions of flat gradient space and never recover. To combat this, many successful modern network models are trained with "leaky" ReLU activation functions, where a small amount of information is allowed through the activation function proportional to the hyperparameter α (Pigou *et al.*, 2018; Viereck *et al.*, 2017; Yin *et al.*, 2017).

On the output layer, the "softmax" activation function was conceived to be used in conjunction to cross-entropy loss (Dunne and Campbell, 1997; Schuster and Paliwal, 1997). This function takes in a vector of activations (or "logits") and transforms them into a valid probability distribution based on the gaussian probability density function.

This method is used overwhelmingly for multi-class detection problems where the likelihood prediction for each class is desired (Jung *et al.*, 2015; Payan and Montana, 2015; Rajpurkar *et al.*, 2017).

It is often said that the fine-tuning of neural network architectures is much like an art, and this sentiment certainly holds true with regards to hyperparameter selection. The long list of hyperparameters used in this thesis is listed in Appendix 1, which exemplifies the extensiveness and intricacy of the hyperparameter tuning process. There are several automated methods that have been implemented to attempt to facilitate the hyperparameter tuning process, such as grid searching (Loshchilov and Hutter, 2016; Nalçakan and Ensari, 2018) and learning curve estimation (Domhan *et al.*, 2015). The selection of hyperparameters is a difficult topic for the data science community to agree on simply because the performance of models with identical hyperparameter configurations deviates greatly for diverse types of data. Typically a data-dependent approach to selection is taken, where the sparsity, quantity, and quality of data are all taken into account to make initial estimates on each hyperparameter (MacKay, 1996, 1999). Other dynamic learning methods such as learning rate scheduling and/or oscillation, as well as time-series attenuation, are applied to adapt the learning rate dynamically to the learning progress of the network (Darken and Moody, 1991; Smith, 2017; Zeiler, 2012).

1.5: Current deep learning frameworks

Open source deep learning frameworks have been published such as PyTorch and TensorFlow (Alphabet, Inc.) that are simplified collections of methods in deep learning

that focus mainly on the automation of the training and validation stages of fitting a learning model to user data. Both PyTorch and TensorFlow can complete the end-to-end machine-learning protocol of data-formatting, model fitting, model testing, and model deployment. They are built primarily for Python developers with the intention of integrating with existing Python libraries to facilitate machine learning in common contexts such as the classification of images, natural language, and sound recognition. The proper handling of data and the deployment of these models however are not standardized. Non-standardized data handling presents room for human error introduced during the pre-processing of the data. This would increase the variability and granularity of the data in undesirable ways that induce a phenomenon called "overfitting", where the learning model gleans from patterns that are specific to the training data but poorly-translatable to outside data. Both deep learning frameworks are capable of automating the intake of some commonly-used databases such as the MNIST and CIFAR databases (Deng, 2012; Krizhevsky and Hinton, 2009; Xiao *et al.*, 2017), thus providing standardized benchmarks that are commonly used to test experimental deep learning architectures. More specific databases however must be programmatically reformatted and normalized into states that are acceptable by neural networks created by these deep learning frameworks. Applications such as Microsoft Excel and Cytoscape (Shannon *et al.*, 2003) have implemented data import templates that offer standardized interfaces between the user's data and the program. Import methods such as these often come accompanied with graphics interfaces and APIs that provide enough flexibility on the parameters while ensuring that the imported data matches with an accepted standard.

Since PyTorch and TensorFlow are libraries developed in Python, the deployment of solutions using them is hindered by the requirement for the Python interpreter. It is possible to use script "freezing" libraries such as pyInstaller (pyinstaller.org) or Glow (github.com/pytorch/glow) that "compile" whole or parts of the Python script by bundling it together with the Python interpreter and any other required files ("dependencies") into a single executable file, thus removing the relatively complex process of installing Python for non-programmers (Abdullah, 2017). This file however carries a large size overhead and multiple performance inefficiencies associated with internal module crosstalk and runtime script parsing. In TensorFlow, the inefficiency of the interpreted Python language is overcome by compiling the learning model into a small program (a "kernel") through the C programming language. In this regard, Python is a middleman between the user and the C language, providing a simplified application programming interface (API) in exchange for performance during data processing, compiling, and communication with the model kernel. The PyTorch and TensorFlow teams recently released a C++ front-end which provides access to a few frequently-used functions directly through C++. The use of the C++ front-end however requires advanced knowledge of several C++ data structures which would be out of the reach of the beginner-intermediate level programmer. A flexible solution rooted in the fast C++ programming language with a gentle learning curve that provides a quick path from data to deployable learning model is currently lacking.

1.6: Big data and next-generation sequencing challenges in clinical oncology

With powerful modern tools and the opportunity for enhanced flexibility in the user experience, we can turn to the numerous powerful data substrates that exist in high-impact research fields to perform the required validations. Clinical oncology has made use of computational tools of increasing power and complexity to attempt to detect, diagnose, and monitor the status of patients during the progression of their disease (Cheng *et al.*, 2015; Cottrell *et al.*, 2014; Guan *et al.*, 2012; Robson *et al.*, 2015). The abundant availability of genome-scale datasets in clinical oncology has presented lucrative opportunities for the formation of high level statistical models of disease (Rhodes & Chinnaiyan, 2005; Hanash *et al.*, 2008; The Cancer Genome Atlas, 2013). Of these tools, machine learning algorithms have become widespread, having found successful applications in many fields of cancer research (Ooi and Tan, 2003; Wei *et al.*, 2004; Libbrecht and Noble, 2015). These flexible tools possess the ability to markedly reduce the time and financial expenses associated with the development of personalized pharmaceutical treatments for the many diverse types of cancer (Bielinski *et al.*, 2014; Lebofsky *et al.*, 2015). Genomics data at the methylome, transcriptome, and proteome levels—colloquially referred to as "omics" data—are difficult to interpret using manual techniques but have been previously interpreted by "prediction analysis of microarrays," (PAM) to detect prognostic gene signatures (Ng *et al.*, 2016; Park *et al.*, 2015; Pongor *et al.*, 2015). PAM represents a collection of machine learning techniques that iteratively apply linear and logistic regression to attempt to internally model the structure of the data, the most popular method being the "nearest shrunken centroids" (NSC) approach

(Leal *et al.*, 2018; Liu *et al.*, 2005; Tibshirani *et al.*, 2002; Wang *et al.*, 2007). NSC is currently one of the state-of-the-art methods in bioinformatics being applied to gene signature detection problems due to their ability to discern polynomial regression coefficients for relatively small numbers of gene contributors (Wang *et al.*, 2007). There have been valid concerns about the applicability of machine learning in cross-dataset comparisons which, at the moment, experience difficulties due to the diverse formatting with regards to data source, storage architecture, and retrieval methods (Goble and Stevens, 2008; Merelli *et al.*, 2014).

As opposed to regression machine learning, neural deep learning methods may show promise in generating generalizable models across datasets due to their ability to self-normalize and discard "noise" data consistently across samples. This has been very recently demonstrated in the consistent detection, and even isolation, of speech from a variety of levels of background noise (Kumar and Florencio, 2016; Qian *et al.*, 2016). These applications apply "very deep" convolutional neural networks that pre-process the data automatically by applying a range of learned transformations before performing deep learning in the fully-connected component (LeCun *et al.*, 2015). Neural networks have already made enormous leaps and bounds in profound applications in clinical oncology, from automated analyses of histology to the detection of deep-level gene signatures (Araújo *et al.*, 2017; Chen *et al.*, 2015; Esteva *et al.*, 2017; Spanhol *et al.*, 2016). Along the tangent of current use cases, deep learning algorithms hold remarkable promise for revolutionizing drug discovery methods in cancer using NGS data. Neural machine learning methods can be well-applied in the macroscopic scope of the cancer disease

where many mechanisms of pathogenesis are poorly understood, and pathological phenomenon are frequently driven by many unseen variables. The current barriers behind the more widespread adoption of neural networks for NGS analysis stem from the sparse availability of sample data, which is currently extremely expensive and time-consuming to collect (Muir *et al.*, 2016; Patel *et al.*, 2016). For a multi-class image classification problem, a typical deep learning neural network requires thousands of samples per class, which makes datasets such as the CIFAR-100 so difficult to learn. Considering that image data presents readily extractable features, this makes image classification a relatively simpler task than prognostic prediction from deep sequencing data. Furthermore, effective deep sequencing predictions by conventional deep learning methods could require many more samples to be robust, unless adaptations are made to fit these models for deep sequencing characteristics such that they become more generalizable. The challenge past that point would be the statistically-valid verification of prediction results from relatively small validations sets, a roadblock which may not be lifted until much more powerful technologies bring down the cost of deep sequencing further.

1.7: Stem-cell like bodies drive heterogeneity of acute myeloid leukemia and disease relapse

The development of intricate algorithms for therapeutic discovery has become a necessary supplement to experimental "wet-lab" validations in cancers research, where patients exhibit high frequencies of post-remission relapse, as is the case in the multiple subclasses of leukemia (Steensma and Tefferi, 2003; Hehlmann *et al.*, 2007; Döhner and

Bloomfield, 2015). Leukemia as a malignancy is diverse in many ways, and subtypes are classified by cell of origin, the presence of pre-existing conditions, cytochemistry, and histology (Arber *et al.*, 2016; Foon, 1986; Vardiman *et al.*, 2009). The complexity of AML, a highly aggressive acquired leukemia, is today widely attributed to the transformation of rare populations of primitive cells in the bone marrow into malignant leukemic counterparts (Hope *et al.*, 2004; Li *et al.*, 2007; Kikushige *et al.*, 2011; Tabatabai and Weller, 2011; Kreso and Dick, 2014). These malignant primitive cells are referred to as cancer stem cells due to their comparable abilities to normal stem cells in the replenishment of more committed cell populations. Cancer stem cells however are aberrant in that the cells they generate are blocked in their differentiation status and remain in a semi-primitive state that is non-functional but competes for the host tissue's energy resources. In leukemia, the presence of a relatively small number of cancer stem cells allows for the disproportionate accumulation of aberrantly differentiated blast cells in the bone marrow niche. This aggregation eventually displaces the healthy hematopoietic stem cell (HSC) population that is responsible for the replenishment of the blood cell supply in circulation (Bonnet and Dick, 1997; Hope *et al.*, 2004; Testa, 2011). These leukemic stem cells (LSCs) are relatively inactive, slowly replicating cell species that, for most of their life cycles, lie quiescent in the bone marrow niche and evade chemical therapeutic treatments. Often this produces the illusion of a successful remission before, as a result of the untargeted LSC, the leukemia awakens at a later timepoint to recapitulate the disease.

Thus, the unique machineries that distinguish these cells from normal primitive bone marrow have been of critical interest (Giustacchini *et al.*, 2017; Jung *et al.*, 2016; Zhang *et al.*, 2016). The gene expression patterns that govern the function and life cycle of LSC cell types are poorly understood and under active investigation with the assistance of next-generation sequencing techniques (NGS) such as RNA-seq and ChIP-seq (Lilljebjörn *et al.*, 2014; Pelish *et al.*, 2015). These techniques are performed on fractions of leukemic blood samples that are differentially enriched for primitive blood cells by their molecular surface markers using fluorescence-activated cell sorting (FACS) (Bernt *et al.*, 2014; Xu *et al.*, 2014). Despite these technologies, the differences in gene expression patterns between LSCs and the normal primitive bone marrow remain difficult to discern with significance due to the low relative proportions of both populations compared to the bulk bone marrow tissue. The variance introduced by cell sparsity is exacerbated by the error margin of the FACS method and the abundant diversity of gene expression patterns and copy numbers (De Magalhães *et al.*, 2010; Treangen and Salzberg, 2012). Due to these barriers, it becomes nearly impossible to capture the wide range of cellular and genetic configurations with significance.

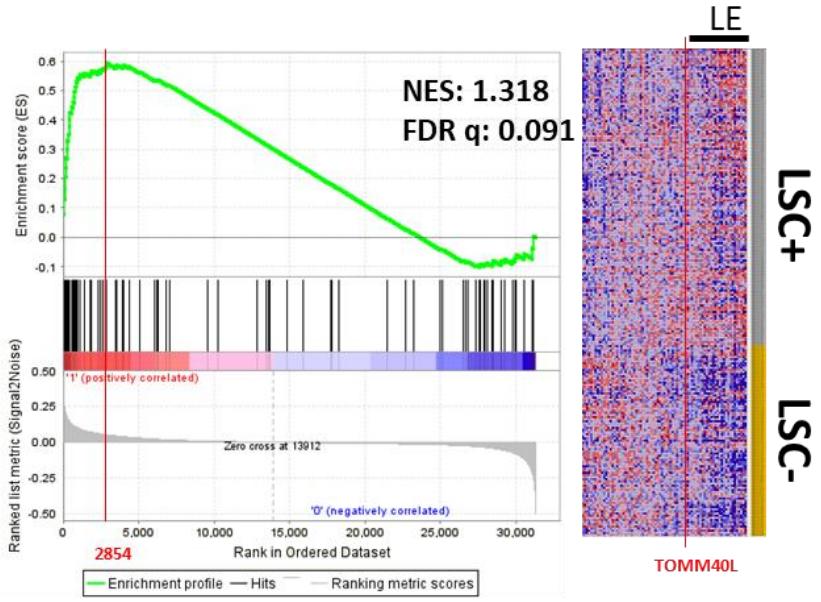
The cell-of-origin question has become highly pertinent due to the discovery that AML tumors, along other cancers, can exhibit "clonal" qualities that are defined by a variety of different genetic characteristics (Li *et al.*, 2016; Young *et al.*, 2016). This knowledge has spurned proactive research into preleukemia and anaemias, which share certain genetic characteristics with leukemia and may explain its mechanisms of transformation (Horiike *et al.*, 1997; Shiozawa *et al.*, 2017; Tiacci *et al.*, 2018). What adds to the complexity of

the AML disease is the differences in gene expression during myelodysplastic states, early leukemia, relapse, and endpoint (Corces *et al.*, 2016; Ho *et al.*, 2016; Kotini *et al.*, 2017; Li *et al.*, 2016). Now when considering that genetic expression profiles may differ by stage, by cell of origin, and by patient genetic variation, the multivariate complexity of genetic dependencies in leukemia becomes apparent. As understandings of AML evolve toward increasing appreciation of its genetic complexity, robust controls, abundant validations, and conservative significance thresholds will become increasingly necessary when applying high-level statistical models for predictive cancer genomics.

1.8: Gene set enrichment analysis (GSEA) for NGS analytics

To address the statistical validity concerns arising from microscale biological comparisons, analysis techniques such as GSEA, which make use of ranked list comparisons in addition to linear and logistic statistical metrics, have become instrumental in validating evidence produced by NGS-based experiments (Kim and Volsky, 2005; Subramanian *et al.*, 2005). It is possible to discern minor differences between samples with similar characteristics using these methods due to the richness of NGS data. Small inflections in gene expression on a single gene are more than likely within the margin of chance, however when differences in many concerted genes arise repetitively across samples the variance becomes less explainable by probabilistic factors. When genes known to interact together are selectively affected between samples, it may indicate that an entire pathway has been affected by the experimental variable. GSEA allows for the visualization of this phenomenon and the assignment of scores to the

likelihood that a data metric is enriching or a group of related genes. To exemplify this, the following GSEA example from Figure 9b has been prepared:

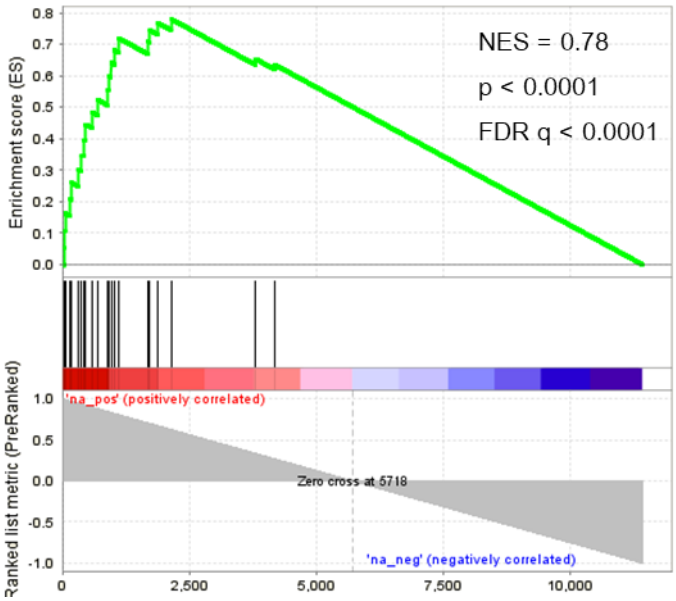


Immediately visible in this plot are several characteristics: a running enrichment score visualized by the fluorescent green line, a gene list metric in sorted order shown at the bottom in grey, and a supplemental gene expression heatmap stratified by phenotype class. This plot visualizes a test of the hypothesis that a specific group of genes can be found at the highest ranks of all genes in a dataset that have been ordered by some specific metric. In this case, the differential expression was calculated for every gene in the microarray dataset across LSC+ and LSC- sorted cell fractions via the signal-to-noise metric, as below:

$$\frac{\bar{x}_{LSC+} - \bar{x}_{LSC-}}{\sigma_{LSC+} + \sigma_{LSC-}}$$

Where $\bar{x}$ represents the population mean and σ represents the population standard deviation. Every gene in the dataset is then ranked from highest signal-to-noise to lowest, which is then analyzed by GSEA to identify patterns at the high ranks of the list. The running enrichment score (RES) is calculated by starting at the top of the ranked list and walking down rank-by-rank to the bottom. The algorithm increments the running ES away from zero when a gene in a gene pathway of interest is encountered, and toward zero when the gene does not belong to the pathway. When enrichment is discovered at either end of the ranked list, the result is a wave-like pattern that crests on the top-left or bottom-right of the RES plot. The maximum deviation from zero made by the running algorithm is returned as the enrichment score (ES) and provides a surface metric for the amount of enrichment encountered during the ranked list walk. This ES score is further normalized to the size of the gene set to provide the normalized ES (NES) which is often reported as the ES result. The significance of the NES is established by calculated by permutation-based statistics, where phenotype labels and gene set order are randomized and the ES is re-calculated (usually 1000 times) to obtain the false-discovery rate (FDR) (Benjamini *et al.*, 2001; Reiner *et al.*, 2003). Other methods such as the nominal p-value (nominal P) and the family-wise error rate (FWER) are also provided as supplementary statistics, however the GSEA developers note that the former is not conservative enough and the latter is too conservative (Subramanian *et al.*, 2005). Generally, a FDR threshold of 0.25 has been established as an acceptable false-positive rate in GSEA (Dinu *et al.*, 2007; Jordan *et al.*, 2016; Pantel *et al.*, 2014).

Though GSEA is often performed using differential expression metrics, the running ES algorithm applies to the identification of the enrichment of any group of labels in a ranked list of background labels. The GSEA application provides an interface to upload "pre-ranked" gene lists in *.rnk file format, allowing researchers to predefine a ranked list using a different metric and then compare it to genetic and molecular pathways to identify if there are any defining characteristics at the extremes of the ranks (Bateman *et al.*, 2014; Murohashi *et al.*, 2010; Musso *et al.*, 2015). This allows for enormous flexibility in signature enrichment detection for virtually any type of metric that can be used to rank a list of labels. In the following GSEA plot taken from Figure 11a, we can see that the ranked list metric appears differently than in the previous example:



There is no expression heatmap to the right of the plot, and the ranked list metric appears to be much more linear than the hyperbolic sinusoid of the signal-to-noise metric graph. This is because the ranked list metric here is simply the rank of each label normalized to

the size of the list and centered at zero. The RES algorithm continues to work and identifies that this gene pathway of interest appears beyond-chance at the high ranks of the pre-ranked list. Since FDR statistics are permutation-based, the validity of this method continues to hold up despite the less canonical approach to GSEA.

Lists of genes involved in concerted pathways that are used for this kind of analysis can be obtained from manually curated online databases. Data repositories such as the Molecular Signatures Database (MSigDB) have curated lists of genes involved in such concerted pathways for use in statistical tests such as GSEA, and for the visualization of pathway workflows using tools like Cytoscape (Shannon *et al.*, 2003; Liberzon *et al.*, 2011). Pathway analyses have been widely accepted as a more fruitful avenue of investigation for drug discovery due to the cascading nature of causality in gene expression (Hennessy *et al.*, 2005; Huang, 1999; Takahashi-Yanaga and Sasaguri, 2007; Thompson and Lyons, 2005). The difficulty in this approach arises when considering that the manually curated spaces account for a miniscule fraction of all the possible molecular interactions in the body. We rely heavily on the previous avenues of investigation to power future investigations, which funnels the research community into limited bottlenecks out of concerns for the safety of their hypothesis. This can be exemplified in the Alzheimer's disease research community, where many reviews have been very recently published in high-impact journals questioning the direct causative nature of β-amyloid in dementia following widespread clinical trial failures (Hardy and De Strooper, 2017; Kametani and Hasegawa, 2018; Makin, 2018). Therefore it remains

important to iteratively re-investigate the experimental data in search of new, more robust leads while using SLR as a mechanism of fact-checking.

1.9: AiDA, a chatbot and NLP-powered solution for enhanced research investigation

A series of algorithms were developed that harbour potential to drastically reduce literature review time by automating the extraction of key decision-making material from large-scale numeric data as well as aggregated unstructured text data. A chatbot interface was constructed in order to facilitate the curation and analytics of the extracted data via the Text-to-Function (T2F) convolutional deep learning system (Figure 1). Furthermore, the platform facilitates SLR by combining NLP technologies and a new cross-dataset ranked-list method in order to increase the throughput of complex multi-factorial analyses. Named the Artificially-intelligent Desktop Assistant (AiDA) platform, it attempts to simplify the user experience by performing activities driven by naturally formatted text requests. The chatbot interface is combined with GUI elements from the open source CVision graphical user interface (GUI) engine (https://github.com/DamianTran/cvision) which is optimized for low-power personal computers and built specifically with internal handles that make automated machine control possible.

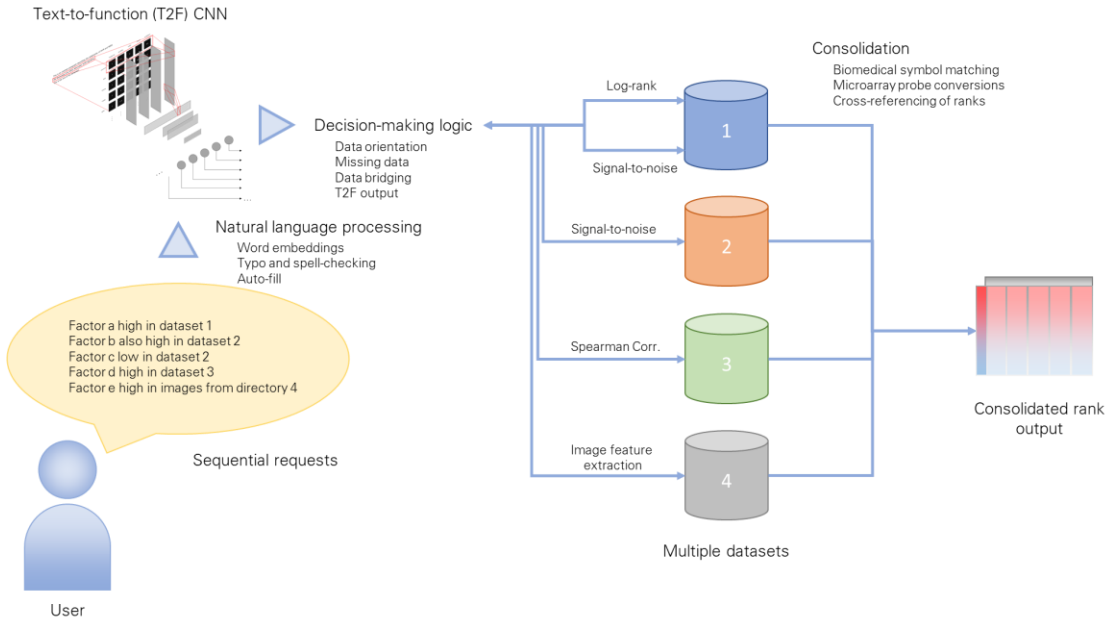


Figure 1. Pipeline flowchart for automated user request processing with AiDA. A user requests the program to discover dataset IDs matching several different factors in several different datasets. This information can be conveyed to the program using informal language, which is processed through natural language algorithms and used to determine the appropriate responses. Responses occur in the form of algorithms applied at large to the user's requested datasets, and result from decisions made by the program that best suit dataset criteria including orientation, sample size, and peripheral datasets in the same directory. The output is a condensed interactive ranked-list that can be easily interpreted for downstream applications.

To benefit researchers and investigators, the AiDA dashboard allows for simpler, less error-prone, and more well-documented analysis of data from many sources. The AiDA chatbot can additionally benefit developers by providing a simple interface to documentation. The chatbot begins with a baseline array of learned responses and can be further trained by creating response pairs using any text editor of choice (ie. MS Word, Excel, Notepad). This allows developers to easily train AiDA to learn responses to

general questions that they anticipate the user to ask. More specific questions by users can be routed to the developer team through the AiDA chatbot email service hosted by Microsoft Outlook, allowing developers to manage customer request chains without requiring the customer to divulge their identity.

The complexity of the proposed work and the specific code architecture required for the speech-to-function system exposed a need for a custom neural network circuitry that would be plastic enough to scale in proportion to the number of inputs, the variability of the data, and the number of required response actions. For scalability and simple deployment the network engine would have to be readily retrainable and preferably reactive to the user's unique profile on their local hardware. Therefore to fulfill requirements for a flexible deep-learning program I created the custom neural network engine using the C and C++ programming languages. This engine produces neural networks that function through a unique combination of range-normalized logarithmic math, matrix transformation, and type-mapped convolution kernels alongside accepted standards for convolutional neural networks. This combination of algorithms empowers neural models with flexible, stretchable perception fields that add additional degrees of freedom to adjust to the function space as well as the intensity of the data. These range-normalized models converge faster and form learning models that generalize better to unseen data.

Considering that it would take an unreasonable amount of time to curate a dataset of sentence-to-action mappings that would be representative of the total number of natural language possibilities, having a learning model that can expand its learning beyond the

small sample it's been exposed to will bring everyday developers closer to mimicking and exceeding the human capacity for sensory learning. The processes of data aggregation and data augmentation are further facilitated with the use of a generalized data construct. This data type adds a layer of processing onto each input data type that allows for generalized handling regardless of the source or format. The result is a much simpler data handling process that expedites the lengthy task of training and validating neural models.

1.10: Summary of Intent

AML is a cancer of the blood with poor prognosis affecting thousands of individuals globally (LLS, 2016). The disease manifests itself through the bulk accumulation of immature leukemic blast cells in patient bone marrow driven by malignant proliferation of leukemic progenitor cells (Lowenberg, 2003). These progenitors are in turn replenished by relatively inactive LSCs that evade modern therapies and contribute to the frequent post-treatment relapses characteristic of AML (Kreso and Dick, 2014). The comparison of functionally validated LSC-enriched fractions extracted from AML peripheral blood to LSC-deficient fractions using RNA microarrays and RNA-seq have presented transcriptome-wide databases of genes differentially expressed across the leukemic hierarchy that form major substrates of the work of this thesis (Eppert *et al.*, 2011; Ng *et al.*, 2016). The analysis of this work is made easier by computational bioinformatic tools through GSEA: massive-scale virtual comparisons of genes having similar biological pathway correlations can isolate a small, concentrated list for a more focused practical purpose (Subramanian *et al.*, 2005).

Since modern methods in next-generation sequencing remain highly costly, the existing AML, LSC, and normal stem cell datasets are amalgamations of samples from diverse origins. Even with the collaboration of multiple international research groups, the coverage of resulting genome atlas projects remains limited in the statistical sense. The complexity and heterogeneity of the AML condition, as well as the rareness of their cells of origin, necessitates a sensitive, robust solution that employs conservative statistics to draw realistic inferences on aggregated data. A series of novel algorithms were implemented in this thesis in order to capitalize on the wealth of open source data currently available for both cancer genomics and stem cell characterization.

This thesis aims to report on the progress of an artificial intelligence platform that makes use of a series of novel algorithms to attempt to automate series matrix data extraction and deep text mining. A custom neural network engine is described that uses a new configuration of activation functions, a cost-inertia hyperparameter optimizer, and convolution of word embeddings to achieve high prediction accuracy from limited, sparse text data. This engine, in sequential combination with several ranked list algorithms, has yielded predictions for novel therapeutic compounds recycled from use cases in other known diseases.

This work seeks to test the hypothesis that drug discovery of reasonable accuracy can be automated end-to-end by a workflow of ranked list feature extraction and deep-learning text mining algorithms.

The null hypothesis in this case would be the inability of the automated drug discovery platform to find tractable leads from the accumulated biomedical evidence beyond chance. To refute this null hypothesis, a list of chemical, enzyme, or functional nucleic acid inhibitors of reasonable impact and actionable value must be produced by the workflow. This would indicate potential for the automation of key rate limiting evidence review steps in the drug discovery process.

Candidates for true positives include:

A. Molecules known to downregulate gene pathways highly expressed in leukemic stem cell samples, but lowly expressed in normal hematopoietic stem cell samples
B. Molecules known to upregulate gene pathways lowly expressed in leukemia stem cell samples, but highly expressed in normal hematopoietic stem cell samples

Thus, the experimental aims of this thesis are to:

1. *Validate the accuracy of novel algorithms postulated for component automation*
2. *Compare the accuracy of novel algorithms against peer-reviewed, standardized benchmarks*
3. *Analyze the predictions of the workflow to discover promising therapeutic compounds for AML.*

The following chapter (2) provides detailed schematics, equations, and tables pertaining to the implementation of several novel algorithms required for the accomplishment of these experimental aims.

Chapter 2: Methods

2.1: CVision graphical user interface library for C++

All interface items appearing in the AiDA dashboard were created in CVision (https://github.com/DamianTran/cvision). CVision was created in the C++ programming language, and is an open-source project to make interface creation more accessible for C++ programmers (schematic illustrated in Figure 2). An application ("CVApp") is instantiated with a media package manager and a runtime loop. For cross-platform portability a "CVView" window is instantiated on the main executable thread and occupies the executable with draw, update, and event handling functions on a framerate-synchronized runtime loop until the view element is closed by the user or an internal function call. All other operations (in this example, the AI runtime loop) are instantiated on other parallel threads and interact with the "CVView" by requesting elements through a Javascript-like API (ie. getElementById("elementTag")).

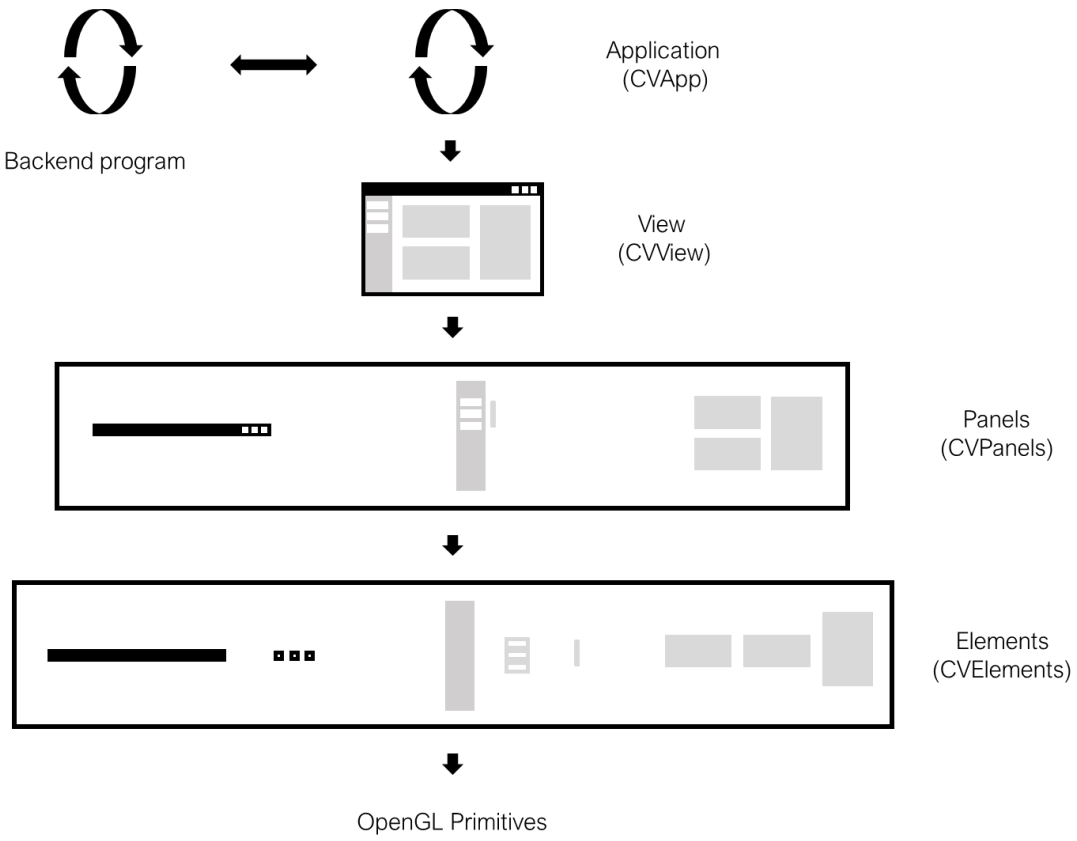


Figure 2. Flowchart of the GUI signal transduction cascade in the CVision engine. Draw, update, and event handling processes are transduced, via virtual class functions inherited from "CVElement", down a hierarchy stemming from the high-level "CVApp" instance. CVApps contain "CVView" instances, which in turn contain "CVPanel" instances that contain "CVElements". CVElements themselves are constructed from OpenGL primitives. User interactions are monitored and accessible through the CVision external API

Drawable CVision elements, or "CVElement" instances, are grouped into CVision panels ("CVPanels") that distribute the draw, update, and event information through cascaded virtual functions. The update hierarchy moves in reverse order of the draw hierarchy, such that elements that are last to be drawn (and thus appear on top) are first to capture operating system events. To create a drawable element, a CVPanel or CVElement item is created using the new operator, parameters are edited such as size, position, color,

outline, animations, and others, and then the item is added to the CVView. At this point, the resources associated with the drawable element are automatically managed such as textures, fonts, primitives, and shaders. For most effective management, CVPanels are created first, such as the "CVBasicViewPanel", "CVListPanel", "CVSwitchPanel", and "CVTogglePanel". CVElements are then created and added to these panels. All items in a panel are drawn and updated collectively, and modifiers are applied to all members equally such as movement, physics, and transitions. Panels may also apply additional properties such as scrolling in the case of the CVListPanel, lateral panning in the case of CVSwitchPanel, and single-panel display in the case of the CVTogglePanel. Colors and fonts can be applied upon creation when CVElement items request theme elements from the CVApp package manager. In this way, all the themes of a CVision application can be managed simply through a simple CVApp control panel. User interactions and timing are tracked by the app and can be accessed programmatically to determine where the user is focusing and what the user is interacting with. This allows for many more powerful automation functions if the CVApp is paired with a backend app that reads the CVApp interaction log to inform changes in app activity based on what the user is currently doing.

2.2: Smith-Waterman local alignment for natural language

String matching in the AiDA NLP engine is scored using the Smith-Waterman algorithm, originally developed by Michael Waterman and Temple Smith, illustrated in Figure 3 (Smith and Waterman, 1981; Waterman and Eggert, 1987; Waterman and Smith, 1986). The Smith-Waterman algorithm is a more computationally expensive alternative to the

often-used Levenstein distance in classic NLP and is frequently used by bioinformaticians for local DNA alignment (Ligowski and Rudnicki, 2009; Pearson, 1991). Like the Levenstein distance, edit events are captured between two candidate strings, however a score is applied differently based on whether the event is a substitution, deletion, or insertion.

The Smith-Waterman algorithm has been recently applied to NLP contexts due to the "gap opening" and "gap extension" penalties that are able to better-discriminate between strings with higher edit distances, or sequential edit events (Gomaa and Fahmy, 2012; Smith *et al.*, 2013). These applications use the algorithm to discover reordered and edited sentence transcripts, but have not been well-explored for the detection of typos and homology between the letters of different words. The source code for the templated implementation of the Smith-Waterman algorithm is a component of the larger open source Hyper library for C++ (https://github.com/DamianTran/hyper).

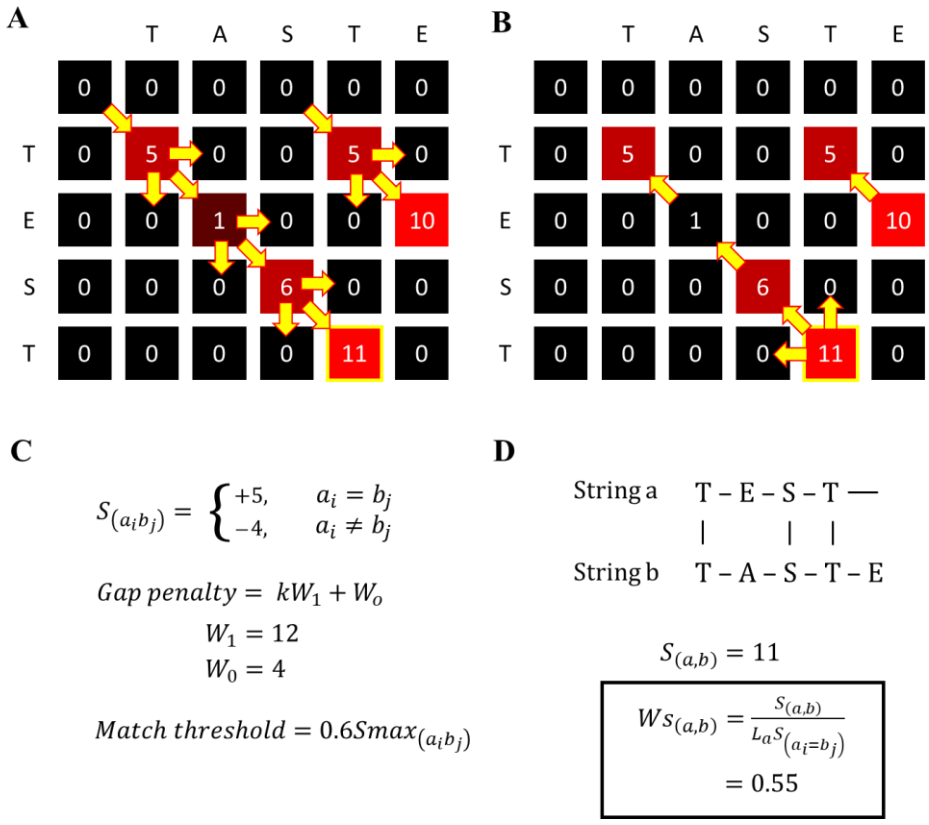


Figure 3. The Hyper C++ implementation of the Smith-Waterman algorithm. The Smith-Waterman algorithm for local string alignment compares two strings, annotated as *a* (TASTE) and *b* (TEST) using a two-step matrix crawl. (A) Step 1 involves the forward calculation of scores by starting at row 0 and looking for initial matches between strings *a* and *b*. Once a match is discovered, the crawling algorithm populates the score matrix based on matches/mismatches (diagonal movement) and gaps (lateral/vertical movement). (B). In Step 2 the alignment is obtained by starting at matrix coordinates with scores greater than the match threshold, and walks backward along the path of highest scores crossing the match threshold. (C). The substitution matrix holds increment values for matches (+5) and mismatches (-4). Gap penalties between strings are incremented based on the gap extension weight (W1) multiplied by size of the gap (k) and added to the gap opening penalty (W0). The match threshold is calculated at the end of step 1 by obtaining the maximum score from the score matrix and multiplying it by the threshold weight of 0.6.

(D). The alignment is obtained by obtaining indices *i* and *j* along the walk for strings *a* and *b*. The final score is normalized to the length of the shortest string (La) and the substitution matrix match score.

The Smith-Waterman score derived herein is a weighted sum of edit events between two strings using a scoring matrix that is one wider than the length of the first string and one taller than the length of the second string (Figure 3a-b). A walk is initiated along the sites of first matches between the two compared strings, and incremented based on the substitution matrix, which is a hyperparameter to the algorithm along with the gap opening and gap extension penalties (Figure 3c). Case-insensitive matches caused an upward increment of the running score equal to the match score, while mismatches caused a decrement of the running score equal to the mismatch score. Here I used a match score of +5, and mismatch score of -4, a gap opening penalty of -12, and a gap extension penalty of -4. It's worth noting that not all implementations of the Smith-Waterman algorithm use the gap opening penalty, however in language contexts where repeated alignment gaps are undesirable the gap opening penalty ensured that these were discriminated against. In testing, the severe gap opening had less of a relative penalty in longer strings with many direct matches, because higher match counts increase the probability that the misalignment was unintentional. When the position of the best alignment is required, it is selected by walking backward from matrix indices containing the highest scores in the score matrix above the test threshold. In all cases where the algorithm was applied to for simple match checking, the backward walk step could be omitted to reduce the computational cost of the algorithm by roughly half. The final score is obtained from the matrix origin of the best alignment normalized to the best

possible score, which is the multiple of the length of the shortest alignment query and the substitution match score. A score of at least 0.5 was required to indicate a positive match between any two strings. Since the algorithm normalization provides relatively higher scores to smaller strings for the same number of edit events, strings of size less than 4 were compared instead by using case-insensitive matching and assigning a score of 1.0 if these exception conditions were met.

2.3: 2D word embedding generation for convolutional deep learning NLP

Raw text was connected to AiDA's deep learning framework by means of feature extraction using a thesaurus of 308 unique word classes covering over 2000 individual words. The thesaurus was maintained in memory and persistent during user interactions, such that novel word associations could be learned and mapped to the various overarching verbal themes. Additional learning was performed by searching key words from each thesaurus category using cURL GET requests to thesaurus.com, and retrieving synonyms from among the HTML document object model (DOM).

Words were mapped to thesaurus categories by parsing paragraphs into sentences based on sentence-terminating punctuation (".", "?" and "!"), and then into words by splitting the white space and all other punctuation marks between them (Figure 4a). Individual words were then soft-matched using the Smith-Waterman algorithm against the thesaurus to identify matching word themes. Each theme was then translated into a point in n-dimensional space, each dimension being an orthogonal metric about the original word that provides the point uniqueness in this virtual space (Series of equations in Figure 4b). A list of dimension names, or metrics, is provided in *Table 1* below.

Metric	**Algorithm**	**Example**
Frequency	Count	Ho, ho, ho (3)
Match	Smith-Waterman	Found/hound (0.8)
Order	$N = 1 \quad 0$ $N > 1 \quad \frac{i}{(N-1)}$	She sells **sea** shells by the sea shore (0.33)
Plurality	Ending in “s”, “ese”, “y”, “i”, “ae”	Many (1)/ single (0)
Emphasis	Bounded by quotes + fraction capitalized	“ELAvl1” (1.5)

Table 1. Orthogonal metrics in embedding generation. The dimension name (metric) on the left is coupled with a brief algorithm description (center) and an example (right). Note that the order score is (0.33) for the bolded example, corresponding to a ratio of 2/6. The C convention of zero-centered memory is applied when calculating the value of i.

As a thought experiment, imagine the words “runner” and “runners” occurring right next to each other in the text, being the only occurrences of those words—in the dimensional space defined by three dimensions of word order, frequency, and match, these points would appear identical. A model relying on the identification of words on these dimensions alone would find difficulty discerning between this and other similar cases (in natural language, this occurs quite often). To distinguish between the similar cases, we can add a fourth dimension of “plurality” that defines the likelihood that these words are plural. A plurality algorithm would be applied to meter these words in a fourth dimension, which would immediately pick out “runner” from “runners.”

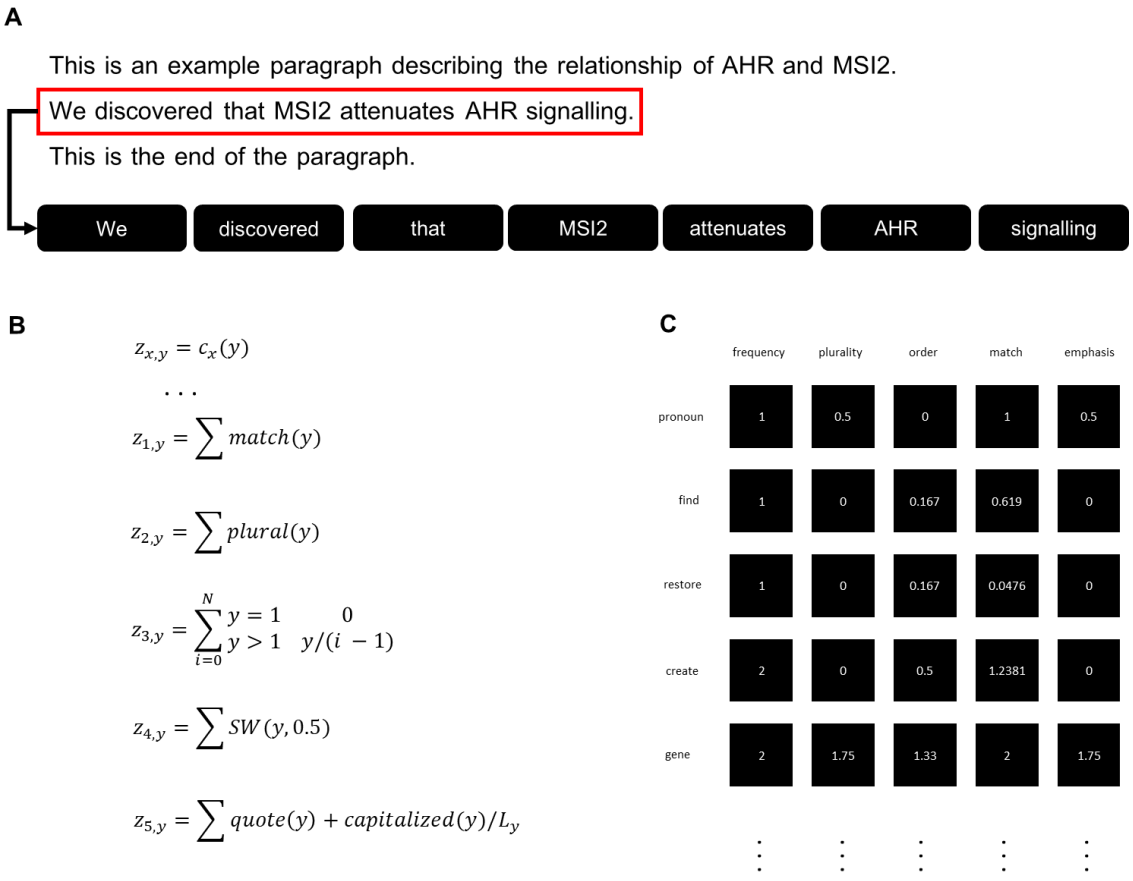


Figure 4. Word embedding generation by n-dimensional flattening. (A). The AiDA parser separates raw text into sentences based on the presence of sentence-delimiting punctuation (“.”, “!”, and “?”). Sentences are then split into separate words based on white-space and all other punctuation marks. (B). Words are cross-referenced to a thesaurus containing 308 word categories to obtain points in the word embedding matrix. A Smith-Waterman match with a score greater than 0.5 triggers the up-dimensioning process, where features are extracted from the matching word to create 5 additional vector dimensions. (C). The dimensions for each thesaurus group are linearized and combined into a 2D word embedding matrix. This matrix is 5 x *n*, where *n* represents the number of thesaurus groups that could be discovered in the sentence transcript.

While n-dimensional vector space is highly useful for providing uniqueness to embedding features in orthogonal dimensions, the data is difficult to access with consistency in computationally efficient ways by constrained MLP systems. Therefore, the multiple dimensions of each word embedding are flattened into a one-dimensional vector, and then laid out orthogonally to each other to create a 2D embedding matrix input (Figure 4c). This embedding matrix, now reduced to two dimensions, conveniently shares characteristics with images such as repeatable features and constrained dimensions. Unlike images however, different text transcripts can have varying "intensities" among the "pixels" of the 2D embedding matrix, and diverse sizes along the dimension harboring the list of thesaurus classes. To achieve consistency between reads, a map of embedding coordinates was retained during the initial network configuration. When each embedding class was encountered in the text, its vector would be assigned the same y-coordinate in the network input layer or feature map on all future reads.

2.4: Custom deep learning engine

A custom deep learning engine was made using the C++ programming language to automate the complete pathway of machine learning development from data handling to deployment (flowchart in Figure 5). Neural network classes were given specialized handling methods for different data formats (ie. binary images, RGB images, plain text). These methods handled the transfer of raw data into a set of node values in the input layer that could be transduced to the remainder of the network architecture using generalized activation functions. Activation functions were implemented for sigmoid, range-sigmoid, rectified linear units (ReLU), exponential linear units (ELU), hyperbolic tangent (tanh)

and softmax equations. Neurons were treated as separate entities with storage members for values, weights, biases, deltas, field depth, and location. Neurons were assembled into layers and managed during forward and reverse propagation differently based on whether they were of convolutional, fully connected, dropout, batch normalization, or pooling types.

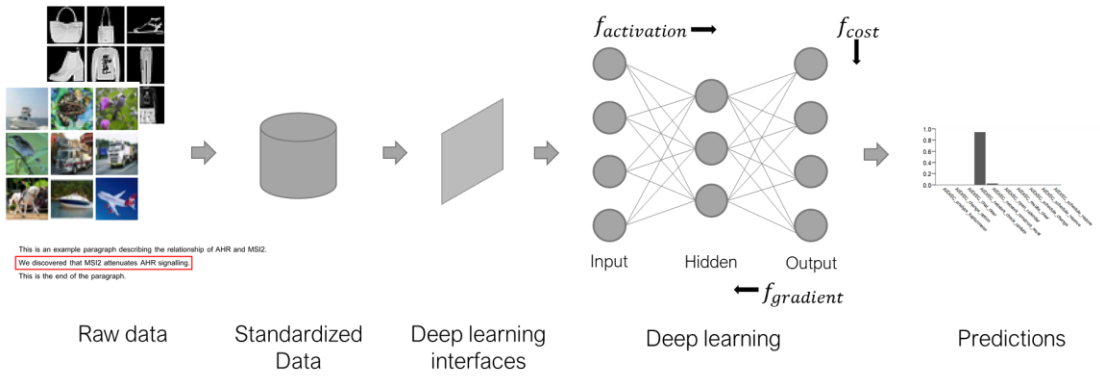


Figure 5. Workflow of the neural network engine. Multiple diverse data formats are handled by the neural network input activation functions. An appropriate filter matrix is created through a pre-processing interface, which is then interpreted by the multilayer perceptron (MLP), or fully-connected layers. The MLP cycle involves feed-forward propagation via the activation function ($f_{activation}$), calculation of the error via the cost function (f_{cost}), and correction of the system via gradient descent optimized by adaptive moment estimation ($f_{gradient}$).

The mathematics of the some custom engine features harbour several deviances from previously described work on feed-forward neural networks and gradient descent (Hecht-Nielsen, 1992; Leshno *et al.*, 1993). Notably, a new self-normalizing sigmoid activation function was used in conjunction to the popular ReLU non-linearity called the range sigmoid (Equation 1). This node-by-node normalization is shares similarities with the

layer-based batch normalization that is applied to a single node at a time, and adaptively boosts or attenuates the incoming signal based on the maximum of the activation components. The self-normalization parameter is the signal gain (μ) which is the inverse of the absolute maximum signal value. This parameter is stored in the neuron as the forward propagation signal passes through it, and enacted upon again during the backpropagation step for use in the calculation of the node delta. For layers below the output layer, the vertical stretch factor *v* was 2.0 and the vertical shift factor *b* was 1.0. This resulted in an output range of $-1.0 < f_{abs(i,l)} < 1.0$. In the output layer, though not applied in this work, the parameters *v* and *b* should be initialized to 1.0 and 0.0 respectively, resulting in an output range of $0.0 < f_{abs(i,l)} < 1.0$. Notably, the form of the range sigmoid is the proper expanded form of the sigmoid function with freedom of motion on all translational parameters.

$$f_{abs} = f(i,l) = \frac{v}{\left(1 + e^{(-\mu a_{i,l})}\right)} + b$$

$$a_{i,l} = \sum_{j=0}^{N_{l-1}} n_{j,l}\beta_{i,l}w_{i,j,l-1}$$

$$\mu_{i,l} = \frac{1}{\max\left(|\delta a_{l-1}|\right)}$$

Equation 1. Self-normalizing sigmoid activation function with transformability. The signal absorption (f_{abs}) for a neuron of index *i* on layer *l* is a relation of the aggregate of input activation functions ($a_{i,l}$) from *N* neurons on layer *l-1*. The outbound signal for a neuron is described as the product of its node

value (*n*), bias (β), and synaptic weight (*w*). The signal gain (μ) applied to the activation function is a horizontal stretch factor that is obtained from the maximum absolute inbound signal component (δa) from the layer below. The function is further transformed by the vertical stretch factor *v* and the vertical shift *b*.

$$f_{cost} = \varepsilon = -\sum_{i=0}^{N_o} e_i \log(o_i) \qquad \frac{d\varepsilon}{dw_i} = e_i - o_i$$

Equation 2. Cross-entropy cost function for the softmax output layer. The cost function (ε, left) is described as the negative sum of natural logarithms obtained from each difference between expected (e_i) and observed (o_i) values for neurons at all indexes *i* in the output layer. Its derivative with respect to the input weight for a single neural output is also described (right).

The cost function (f_{cost}) is formulated in Equation 2 and represents the cross-entropy loss of the system at any given trial. The derivative of this function for each output neuron is the difference between the observed and expected value and is the origin signal propagated from each output neuron toward the input layer during backpropagation. The The Nadam optimizer was used which calculates first and second order estimates on the running averages for all trainable parameters, and additionally applies Nesterov momentum to the first moment (Dozat, 2016). During testing, it was found for this use case that the additional momentum coefficient helped propel the model through long segments of flat gradient surfaces, brought on mostly by the sparsity of smaller transcripts.

$$f_{gradient} = \frac{\partial \varepsilon}{\partial w_{i,j,l}} = \frac{\partial \varepsilon}{\partial n_{i,l}} * \frac{\partial n_{i,l}}{\partial a_{i,l}} * \frac{\partial a_{i,l}}{\partial w_{i,j,l}}$$

$$\Delta w_{t+1} = \frac{1}{B}\sum_{1}^{B} \eta \frac{\partial \varepsilon}{\partial w_t} + m \frac{\partial \varepsilon}{\partial w_{t-1}}$$

$$\delta_{i,l} = \frac{\partial \varepsilon}{\partial n_{i,l+1}} * \frac{\partial n_{i,l+1}}{\partial a_{i,l+1}}$$

$$\frac{\partial \varepsilon}{\partial n_{i,l}} = \begin{cases} e_{i,l+1} - w_{i,l+1} & l = L \\ \sum_{i=0}^{N_l+1} w_{i,l+1}\delta_{i,l+1} & l < L \end{cases}$$

$$\frac{\partial n_{i,l}}{\partial a_{i,l}} = \frac{\partial f_{activation}}{\partial a_{i,l}} = \mu_{i,l}\left(n_{i,l} + b\right)\left(v - \left(n_{i,l} + b\right)\right)$$

$$\frac{\partial a_{i,l}}{\partial w_{i,j,l}} = w_{i,j,l}$$

Equation 3. Reinforcement gradient function for synaptic weight correction in stochastic gradient descent with momentum (SGD-M). The gradient function ($f_{gradient}$) represents the partial derivative of the cost function (ε) with respect to a synaptic connection weight ($w_{i,j,l}$) between a neuron at index *i* on layer *l* and a neuron at index *j* on layer *l-1*. Using the chain rule, the derivative is supplied down the network and can be split into three differentials: the derivative of the cost function with respect to the output value of neuron the neuron on layer *l* ($n_{i,l}$), the derivative of the output value of neuron ($n_{i,l}$) with respect to the incoming activation ($a_{i,l}$), and the derivative of the input signal with respect to the synaptic connection weight. On the output layer (*l* = number of layers, *L*) the derivative of the cost function with respect to the outbound signal is simply the derivative of the cost function alone. On layers below it, the derivative is the sum of outgoing signal strengths multiplied by the error delta (δ) of the neurons they connect to. The derivative of the output node value with respect to the activation input is the derivative of the activation function with respect to the activation. Finally, the derivative of the activation with respect to the input is the freely-variable neuron weight. The synapse weight is finally modified by the result of the gradient calculation at read point *t*, limited by the learn rate (η) and amplified by the last gradient at read point *t-1* modulated by the neuron momentum (*m*).

For baseline stochastic gradient descent with momentum (SGD-M), the mechanisms of backpropagation are delineated in Equation 3. The complex derivative of error with respect to each trainable parameter (*w*) is split by chain rule into three distinct differential equations representing:

1. The rate of change of the cost function w.r.t. the node value: $\frac{d\varepsilon}{dn}$
2. The rate of change of the node value w.r.t. its activation: $\frac{dn}{da}$
3. The rate of change of the activation w.r.t. the synapse weight: $\frac{da}{dw}$

The gradient function is computed at each read point *t*, and error deltas (δ) are computed based on the components of the chain rule separation that are not free to vary with respect to each synapse weight (i.e. the first two components in Equation 3, line 2). Error deltas are aggregated in the neurons for a number of read points equalling the read batch size (B). When the number of reads equalling the read batch size have passed, the aggregated deltas are used to modify all synapse weights at once by the average of accumulated deltas. The amount of modification is a fraction of the aggregated deltas proportional to the network learn rate hyperparameter (η), which was initialized at 0.001 for all experiments and changes dynamically according to the convergence of the system using the hyperparameter cost-inertia system. In the example in Equation 3, a relatively simpler prelude to momentum is provided, which acts to push the system through shallow local minima into deeper, more global minima of the cost function surface.

For all deep learning applications in this work, the Nadam optimizer was applied. This optimizer is a variant of the Adam optimizer, the name of which was derived from the short form of "adaptive moment estimation" (Kingma and Ba, 2014). The Adam optimizer is currently considered state-of-the-art and used in many winning contributions to Kaggle for complex image recognition tasks due to its computational efficiency and the collectively reported fast convergence rates (Bello *et al.*, 2017; Raissi, 2018; Richardson, 2018; Salehinejad *et al.*, 2018). There have been mathematical rebuttals refuting the ability of Adam to find true global minima (Keskar and Socher, 2017; Reddi *et al.*, 2019), and in testing performed during the development of this project it was observed that the Adam optimizer presented a slight overfit of about 1% (±0.5%) compared to SGD-M. Thus, Nadam was adapted to the custom engine created for this work to supply inertia to the nascent models and caused them to perform on-par with SGD-M results. The application of the Adam and Nadam optimizers, and slight differences between the first order estimates are exemplified in Equation 4 below.

Adam

$$m_{t+1} = \beta_1 m_t + \delta_t(1-\beta_1)$$
$$v_{t+1} = \beta_2 v_t + {\delta_t}^2(1-\beta_2)$$

Nadam

$$m_{t+1} = \beta_1 m_t + \delta_t(1-\beta_1)$$
$$v_{t+1} = \beta_2 v_t + {\delta_t}^2(1-\beta_2) + (1-\beta_2) * \frac{\delta_t}{1-\beta_1^T}$$

Both

$$\beta_1^T = 1 - \beta_1^{t+1} \qquad \varepsilon = 1 * 10^{-8}$$
$$\beta_2^T = 1 - \beta_2^{t+1}$$

$$\widehat{m}_t = \frac{m_1}{(1-\beta_1^T)}$$
$$\hat{v}_t = \frac{v_1}{(1-\beta_2^T)}$$
$$w_t = w_{t-1} + \eta \frac{\widehat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}$$

Equation 4. Summary of moment estimates and update function for Adam and Nadam optimizers.

Both optimizers obtain the first (*m*) and second (*v*) order estimates at time *t+1* by applying a corrective bias respective to each order (β_i) to the running average for each at former time *t* and incrementing by the delta attenuated by the inverse of the bias. The second order estimate uses the square of the delta, effectively attempting to predict "ahead" of the current gradient by re-applying the delta rule. In the Nadam case, Nesterov momentum is applied to the second order estimate to provide inertia to the forward prediction. In both cases, time-corrected biases β_i^T respective to each order are calculated to account for initialization bias. The first and second order estimates are thus corrected by dividing by the inverse of these time-corrected parameters. The trainable parameter *w* is then incremented by the fraction of the corrected first over moment over the square root of the second order moment. The epsilon parameter ε is virtually always 10^{-8} and exists to maintain the stability of the system during low second moment estimates.

2.5: Cost-inertia hyperparameter tuning for maintenance of neural convergence

In order to maintain stable convergence of the system and improve convergence during long training runs, a new parallel hyperparameter tuning system was implemented. This system, called the "cost-inertia" hyperparameter tuning method, was initialized on a parallel thread to the main neural network scan. Its purpose was to monitor the progress of the network as it converged by taking regular reads of the output loss in order to make informed predictions about how the learning rate, weight decay, and gradient noise hyperparameters should be tuned. The cost inertia dampening (ζ) is applied at every time point to attenuate or amplify each hyperparameter based on the rate of convergence. The "inertia" property is conferred due to the response cycle of the dampening coefficient to the first order cost estimate.

The cost-inertia system took reads of the running cross-entropy loss average at a pre-determined frequency (24 samples per second) and stored these in a 5-second buffer, a temporary storage technique often used in voice activation applications. The first order derivative of the loss function was then estimated by using least squares regression on the last 5 seconds of the running loss average in order to calculate the cost-inertia dampening (ζ) as follows:

$$\zeta = 1.0 + \left|\log_{10}\left|\frac{\Delta\varepsilon_t}{\Delta\varepsilon_o}\right| * 1.07^{\frac{t}{N}}\right| \qquad \Delta\varepsilon = \frac{d\varepsilon}{dt}$$

Equation 5. Formula for calculation of the cost-inertia dampening coefficient. The cost-inertia dampening (ζ) is necessarily always larger than 1, and is inverse to the state of hyperparameters η, γ, and ψ

at time *t*. The base-ten logarithm of the ratio of the first order estimate at time *t* ($\Delta\varepsilon_t$) to the initial first order estimate ($\Delta\varepsilon_o$) is multiplied by the constant 1.07 raised to the power of the current epoch (*N*) fraction, then added to the minimum range of ζ.

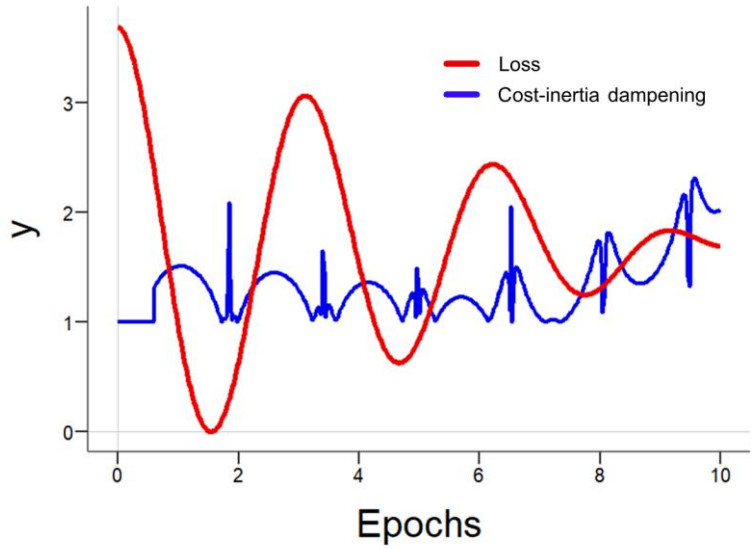


Figure 6. Effect of cost-inertia dampening on oscillating network loss. The response of the cost-inertia dampening coefficient (ζ, blue) is compared to a hypothetical oscillating network loss (red) over 10 epochs, each with 100 samples. A read rate of 96 samples per second is assumed, with a first order estimation buffer of 120 samples. The oscillation was modelled by applying $\cos\left(\frac{x}{50}\right)$ for vector *x* [1, 2, 3, …, 1000], performing parallel vector multiplication with vector *y* [3, 2.997, 2.994, …, 0] and then scalar addition to the minimum of vector *x* * *y*.

The effect of this dampening is visualized in Figure 6, where the cost-inertia dampening increases drastically in regions where the loss begins to change directions and decreases during long stretches of constant motion. Effectively, the dampening coefficient serves to promote consistency in the motion of the loss function by preventing rapid changes in direction. Each hyperparameter is updated by division of the initial hyperparameter value

by the dampening, where larger values of ζ result in inversely smaller values of η, ψ, and θ relative to their initial values at the beginning of training.

$$\eta_t = \frac{\eta_o}{\zeta} \quad \psi_t = \frac{\psi_o}{\zeta} \quad \theta_t = \frac{\theta_o}{\zeta}$$

2.6: Text-to-function deep learning architecture

Text was translated into automated function calls through a connected system involving the AiDA NLP algorithms (described in 2.2, 2.3), the n-dimensional word embedding algorithm (described in 2.3), and a custom convolutional neural network architecture (Figure 8). Paragraphs were split into sentences and then converted into a dictionary-labelled 2D word embedding matrix. From here, two different architectures were tested: an isolated MLP unit with 2 hidden layers and relatively high complexity (Figure 7), and a convolutional neural network (CNN) with a relatively simpler MLP unit also containing 2 hidden layers (Figure 7).

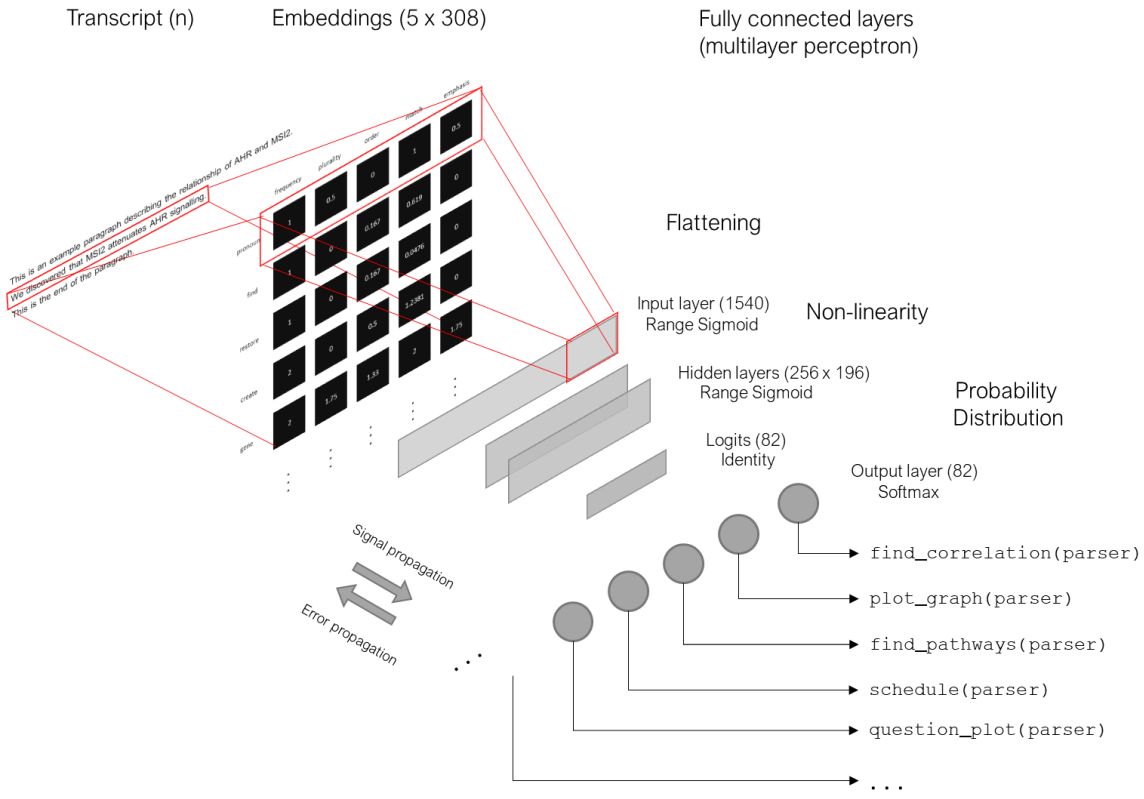


Figure 7. Text-to-function deep learning multilayer perceptron architecture. Text was separated by white space and punctuation via the parser and scored in 5 dimensions to create the embedding vector space. The embeddings were then flattened into a single dimension, each dimensional measurement for each point was interpreted by a single neuron. Before training, an embedding index was configured for the input layer to properly map indices of embeddings with different size/y-dimensions order consistently to the proper input neuron. Input signals were fed through the hidden layer via ReLU and range sigmoid activation, through the softmax layer, and finally into the output layer where a the model's inference of the text could be translated into a gaussian probability distribution via the softmax layer. The final action was selected based on the index of the maximum output probability.

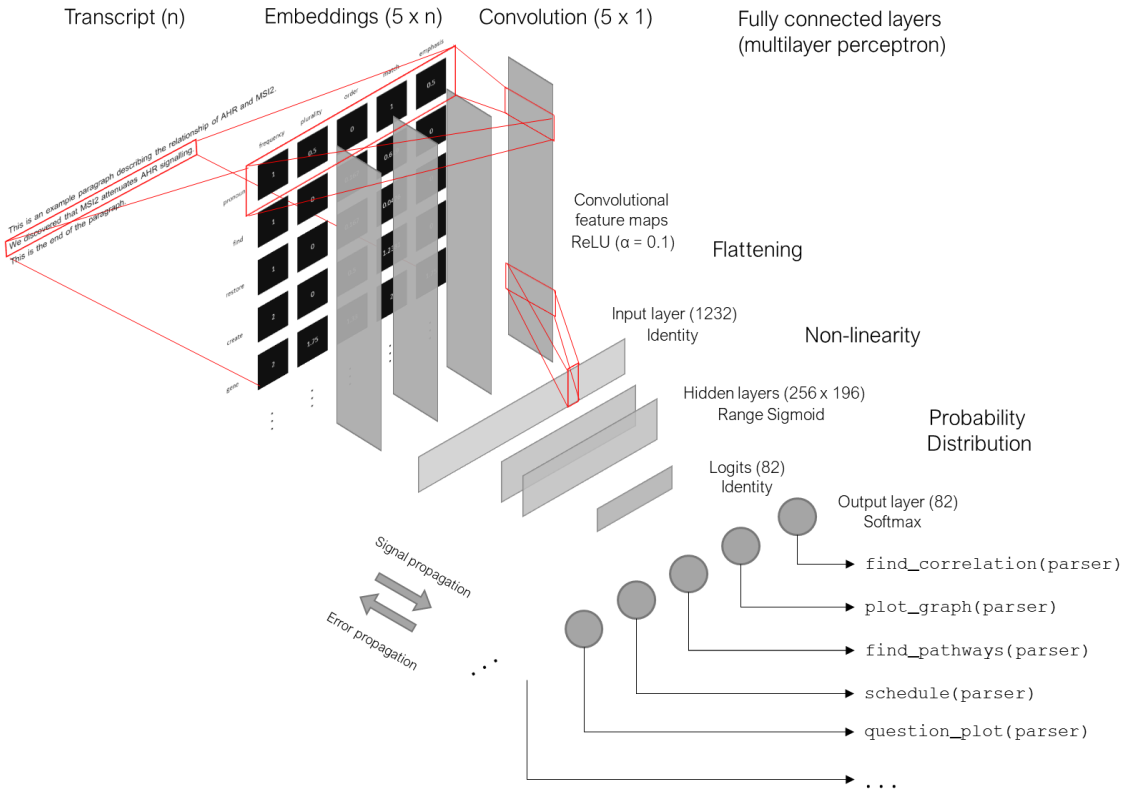


Figure 8. Text-to-function deep learning convolutional architecture. Text was separated by white space and punctuation via the parser and scored in 5 dimensions to create the embedding vector space. The embeddings were then flattened into 2 dimensions during the convolutional process, whereby all 5 dimensions where convoluted by a 1D filter. The filter layer was flattened into the input layer of the multilayer perceptron. An embedding index was configured for the convolutional filter maps before training to consistently map indices of embeddings with different size/y-dimensions order to the proper coordinate. Input signals were fed through the hidden layer via ReLU and range sigmoid activation, through the softmax layer, and finally into the output layer where a the model's inference of the text could be translated into a gaussian probability distribution via the softmax layer. The final action was selected based on the index of the maximum output probability.

The embedding matrix was then convoluted by 4 [5 x 1] linear kernels to compress the input matrix into a 1D feature map, which was fed into the MLP. The output layer was

activated via the softmax method to obtain a probability distribution summing to 1.0, that represented predictions on the likelihood that each function corresponded to the input text transcript. The index of the maximum score in the output vector was then used to select the predicted action function from the labelled output neurons.

T2F CNN training was performed over 75 epochs over 25 permutations, where each time the models were completely reconstructed and weights were randomly initialized via random sampling of the gaussian normal distribution with mean 0 and standard deviation of $\sqrt{\frac{2}{N_l+N_{l-1}}}$, where N represents the number of neurons on layer l. Hyperparameters used were: learning rate (η) = 0.01, weight decay (γ) = 0.01, gradient noise (ψ) = 0.5, gradient clipping (ρ) = ±1.0, backpropagation loss threshold (θ) = 0.05, batch size (B) = 4. The Nadam optimizer (Adam with Nesterov momentum) briefly described in *Methods 2.4* was applied at each batch. Cost-inertia tuning was applied in parallel during the entirety of each run, reinitialized at the beginning of each permutation to $\zeta = 1.0$.

A custom T2F dataset of 1265 manually paired transcript-function names consisting of 82 unique function labels was used (included in Supplementary File S3). 81 function labels were assigned 15 transcript examples each, with the exception of the "non_function" group which was assigned 50 labels. Blind random splitting of the total set was performed at a 73%/27% ratio for every training permutation, where the training set was assigned 11 random selections of each label (37 from the "non_function" group) and the test set was assigned the remainder. Therefore, each training run consisted of 928 text-function pairs, and each test run consisted of 337 pairs.

2.7: Hierarchical search tree for complex, resource-intensive lookup tasks

There are many resource-heavy look-up tasks that the AiDA chatbot is often required to do, such as filesystem searches, named entity searches, and value key conversions. I found that as the number of complex lookups increased with the development of the parser structure, the performance of the runtime AI suffered. To combat this, I created a templated lookup hierarchy structure, which assembles a dendrogram of alphanumeric branches according to the position of characters in a string type, and stores an immutable memory connection (a "reference") to the original item. As a result, the time-complexity of searches became virtually linear for all sizes of lookup keys. This was because a search for string "abcd" would begin at the base of the hierarchical tree with words only beginning with the letter "a", then would move to the second tree under branch a1 containing only words with a first letter "a" and a second letter "b", and so on. The number of required match checks in the worst-case scenario drops from hundreds of thousands to a few hundred. The assembly and modification of the search structure was made more efficient time-wise and memory-wise by storing references instead of copies of the original strings, allowing many of these structures to be internally constructed by AiDA without running out of RAM space.

2.8: Ranked list algorithm for multifactorial cross-dataset consolidation of named features

In order to address the issue of data heterogeneity across diverse datasets, a ranked list algorithm was implemented. The ranked list algorithm extracts features from individual datasets and performs all comparisons between those features in the isolated environment

of each dataset. This process produces a ranked list for each input dataset which can then be used to consolidate results across sets by comparing the ranks of each named feature. The process of rank comparison discards artifacts that may confound any predictions made between algorithms on the low level, such as sample sizes, qualities, and intensities, as well as hardware such as sequencing platforms.

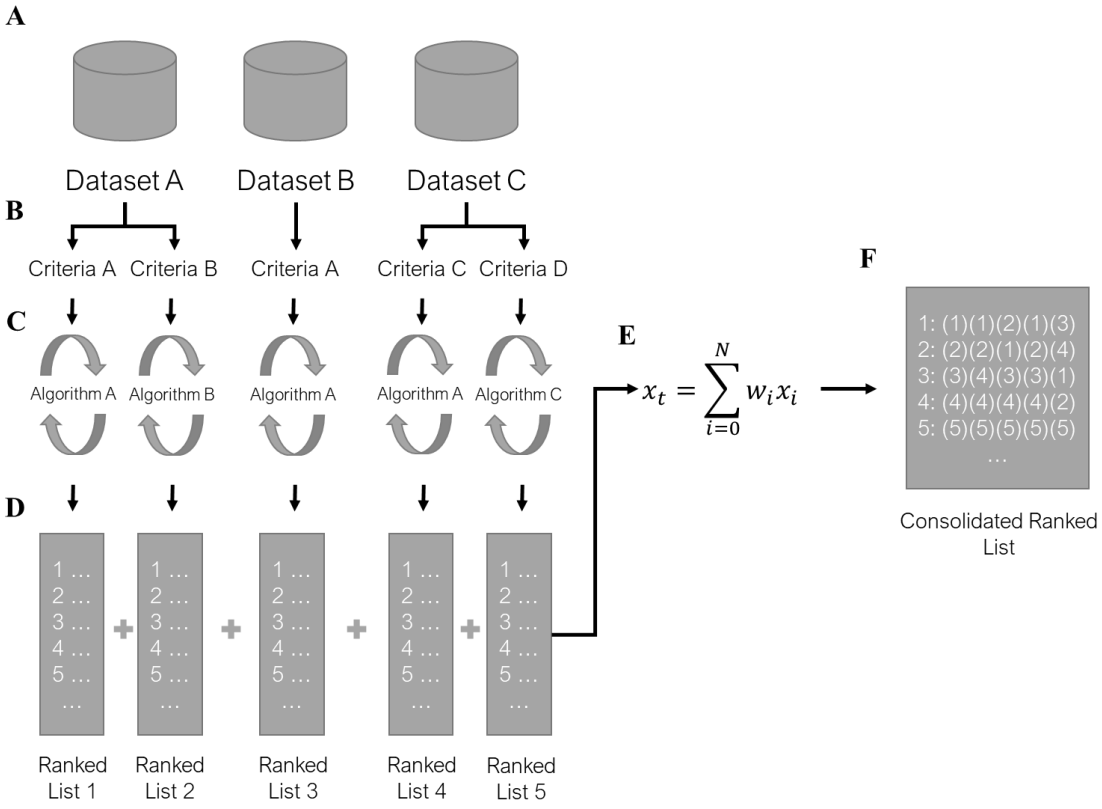


Figure 9. The AiDA ranked list algorithm. Shown above is a flowchart describing the process of multi-dataset feature extraction, criteria ranking, and weighted consolidation into final combined ranked lists. (A). The ranked list algorithm ingests data from multiple source that can be normalized in different ways and/or measured using different metrics/platforms. (B). Criteria are selected from each dataset. The appropriate algorithm for feature extraction is selected based on the discrete/continuous nature of the criteria, and other key words in the user's request. (C). Once the appropriate algorithm is selected

according to the requested criteria, it is used to extract features iteratively along the most meaningful orientation of each respective dataset. (D). Extracted features are ranked from 1 to *N* based on descending order of their algorithm score for each criteria. (E). The final weighted ranks (x_i) for each ranked list feature at index *i* is calculated for all features that can be matched by their name across criteria. Named entities that do not appear in all ranked lists are dropped from consideration in the final consolidated ranked list. (F). The consolidated ranked list is produced, and consists of combined ranks, annotated by the original dataset, criterion, and algorithm that was used to produce each rank component.

The rationale behind the ranked list algorithm (Figure 9) was that if named features consistently ranked high across several criteria and several datasets, they would have statistical grounds to be selected as promising candidates for future study. The consolidated ranked list output that captures these candidates is a weighted sum of ranked named features that were calculated from the addition of multiple vectors of other ordered sub-lists. The power of the ranked list method arises from its ability to consolidate extracted named features across datasets of heterogenous origin, normalization, and metrics, and between criteria with heterogenous feature extraction algorithms. The output is a pure representation of candidates based on the ranked degree to which they match all requested criteria in all independent datasets. Feature extraction as mentioned in the context of the ranked list algorithm is defined as the process of applying algorithms iteratively for all detectible named entities along the axis orthogonal to the alignment of the dataset labels. If the labels are row-oriented, the algorithm will run column-wise and associate individual named entities with rows, and vice versa. When a user requests that AiDA find features of a dataset, (s)he is creating a feature list "building block" that can be useful by itself but even more so as part of a consolidated ranked list.

The scores for each extracted ranked list are determined based on the type of question asked. A question referring to discrete qualities of the data such as "find correlations to high risk" will cause AiDA to use the signal-to-noise ratio to rank all named features in the data according to their enrichment in "high risk"-labelled data. On the other hand a question referring to continuous qualities of the data such as "find correlations to gene X" will cause AiDA to use the spearman correlation to assess the numeric correlation of all named features in the data to "gene X." Elements of the NLP engine described in *Methods 2.2* and *2.3* are used to find the best match to the user's request if no direct label matches are found. Similar string matches and synonyms are also considered when searching for data labels of interest and named features. A table of implemented feature extraction algorithms is shown in Table 1.

Example Criterion	**Label Type**	**Algorithms**
"Find correlations to gene X"	Continuous	1. Spearman correlation 2. Pearson correlation
"What's associated with X category patients?"	Discrete	1. Signal-to-noise ratio 2. Point-biserial correlation
"Find correlations to overall/disease-free survival"	Continuous	1. Kaplan-Meier survival curve 2. Log-rank test 3. Hazard ratio
"What are the lowest X values in the data?"	Continuous	**Depending on X:** 1. Median (Default if no X provided) 2. Mean 3. Mode 4. Signal-to-Noise

Table 2. Criteria match table for implemented feature extraction algorithms. Algorithms represent potential responses to requested example criterion. The default response algorithm is the first in the list in each cell, followed by other algorithms that have been implemented and tested but did not perform as well.

2.9: Automated data bridging

To facilitate the handling of fragmented datasets, or datasets with separated data matrix and annotations, I implemented a data bridge protocol that can create "portals" between datasets. These portals have an "outbound" and "inbound" side, typically mapping to the data matrix and associated annotation set, respectively. Portals are situated upon "staging lines," which represent the line of features that directly map to features under the opposite portal on the matching dataset. All searches initiated on datasets with an outbound portal will continue inline with the dataset holding a matching inbound portal. The linear field emerging with field lines orthogonal to the outbound staging line is transferred to the inbound portal and field lines are transformed orthogonal to the orientation of the inbound staging line. The orientation of data of interest in a dataset is determined by assessing whether the x- or y-distribution of their point cluster is more diffuse. A higher y-diffusion (or higher standard deviation of y-coordinates) represents data that is column-oriented, while conversely a higher x-diffusion indicates that the data points are row-oriented (Figure 10a). This orientation is assessed whenever an automated inference needs to be made about what the user is requesting when they ask for information relating to a data label.

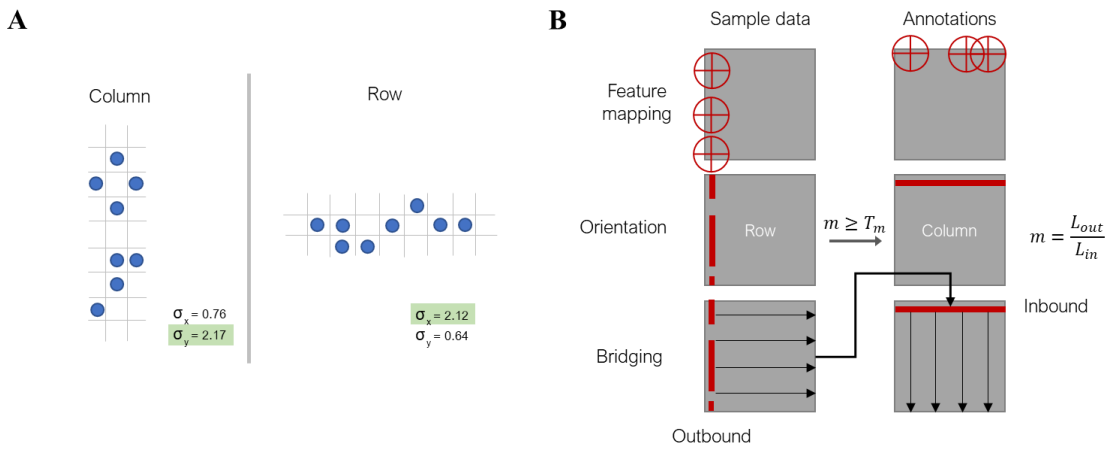


Figure 10. Automated label orientation and data bridging. (A). The standard deviation (σ) of x and y coordinates in a match set are analyzed to determine the orientation of a coordinate cluster. A higher standard deviation of y-coordinates indicates row-oriented data, while a higher standard deviation of x-coordinates indicates column-oriented data. (B). The data bridging protocol begins when two data files are compared for matching features. The algorithm walks down rows and columns to identify regions of high similarity (such as sample IDs) in order to discover staging lines for the "outbound" and "inbound" bridge portals. The orientation algorithm in (A) is applied to determine the orientation of the outbound and inbound staging lines. If an orientation can be determined for both regions, the match score *m* is calculated based on the outbound staging line length (L_{out}) normalized to the inbound staging line length (L_{in}) to determine if the candidate datasets pass the bridge threshold (T_m, initially 0.6). If *m* number of matching labels discovered is greater than the bridge threshold then a data bridge is instantiated and stored in memory for future reference. Further search operations on the dataset with the outbound portal will always include an inline search of the dataset containing the matching inbound portal.

The candidacy of two datasets for data bridge formation is assessed by searching in a cross-hatch iteration pattern across both datasets for highly similar label features using the NLP methods described in *Methods 2.2* and *2.3*. Highly similar matches across both datasets are tallied, and the orientation of matching coordinates is assessed as a cluster to

determine their distribution, and therefore their relevant orientation within the dataset. If an orientation can be determined for both candidates, the mode of coordinates in the orthogonal dimension to the orientation is selected as the site for the staging line. A match score is assigned to both the outbound and inbound candidates that is equal to the ratio of the outbound staging line size to the inbound staging line size (Figure 10b). If the match score is greater than the match threshold, which is initially 0.6, then a bridge is formed between the two candidate datasets.

2.10: The "Spider" web crawler bot

Web scraping and web crawling tasks were accomplished using the Hyper C++ Spider web bot (https://github.com/DamianTran/hyper). The bot was used for domain-specific searches by "placing" it on a key domain search URL (such as a page with a searchbar) using the navigator function `go_to(URL + extension).` The Spider uses an HTML tree parser to extract the document object model (DOM) of online *.html sources downloaded using the cURL C API, libcurl (https://curl.haxx.se/libcurl/c/). The Spider navigator keeps track of where the bot has been, and what links are left to navigate at depth *n* from the origin. Body text, links, and link attributes are extracted from the DOM to allow easy navigation and extraction of web text where applicable (<p>, <h>, and <a> classes). The bot is equipped with algorithms from *Methods 2.2* and *2.3* to allow it to extract key terms from web pages that come proximal to web links, or that occur within the body of a web link URL. Altogether, these tools allow the developer to jump into programmatic web surfing in order to automate data extraction. The web bot was

specifically implemented in the AiDA AI for tasks such as thesaurus lookups and UCSC gene sequence fetching.

2.11: Automated testing methods

The T2F system was validated by permutating each model 25 times using a randomly-segregated 73%/27% train/test ratio. Gradient checking was performed on a small subset of weights and biases to ensure that backpropagation deltas were being computed properly. In order to check I/O consistency of the neural network engine, trained models were fit and validated, saved to the disk, reloaded from scratch, and then validated once again to ensure that the same result was obtained. Trainable parameters were checked for identity iteratively before and after reloading to ensure that all information was transferred to the drive and back into RAM without corruption. The model graphs for training/testing were completely randomized, and reads were performed in no particular order to avoid any sequential biasing. New algorithm functions were permuted with random numbers from $-1*10^9$ to $1*10^9$, as well as NaN, 0, and infinite values to test for crashes. Extensive exception handling and GDB debugging were used to identify program malfunctions and code errors in the source files.

2.12: Datasets consumed by the ranked list algorithm

Normalized datasets used in the development of clinical and biochemical prediction models were obtained from the NCBI Gene Expression Omnibus (GEO) database and cBIOportal. Clinical survival data was analyzed from the LAML-TCGA provisional dataset (The Cancer Genome Atlas AML, n = 173) and GSE12417 (n = 86). In LAML-

TCGA, patients with all karyotypes, mutations, and treatment regimens with RNAseq (V2 RSEM) data were included. Pre-normalized RPKM values, as well as clinical labels and whole-genome mutation information were obtained from cBIOportal for LAML-TCGA. RPKM values were converted to log2 RPKM in R. GSE12417 contained Affymetrix Human Genome U133 Plus 2.0 microarray data for 86 samples (79 bone marrow, 7 peripheral blood); all 79 patients were afflicted with normal karyotype AML. Gene expression across the myeloid arm of the hematopoietic hierarchy was analyzed using GSE42519 which included 34 sorted bone marrow samples (4 HSC, 2 MPP, 3 CMP, 5 GMP, 2 MEP, 3 early promyelocytes, 3 late promyelocytes, 2 myelocytes, 3 metamyelocytes, 4 band cells, 3 polymorphonuclear cells) sequenced by Affymetrix Human Genome U133 Plus 2.0 microarray. Differential expression in LSC-containing leukemic blood was assessed from a cohort of 78 untreated AML patients in GSE76009, consisting of 227 T-cell depleted mononuclear cell samples sequenced by Illumina HT-12 v4.0 expression beadchip (138 LSC+/89 LSC-). All datasets downloaded from GEO (GSE12417, GSE42519, GSE76009) were obtained in series data matrix (*.txt) formats containing $\log_2$-normalized probe intensity values. To enhance program portability, temporary probe conversions to the maximally-responding probe for each gene were performed as required.

2.13: Linear statistical methods

Coefficient of determination (R^2) values predicting gene co-expression were calculated using the least squares regression method for the spearman correlation. P-values for differences in patient cohort outcomes for Kaplan-Meier survival curves were calculated

using the log-rank test method with right censorship. Significances between sample groups selected by dataset labels were determined by calculating the signal-to-noise ratio of the selected group against the background of the remainder. Verbal enrichments were determined to be significant by calculating the Fisher exact p-value for a 2x2 contingency table as follows:

Subset with term *a*	Subset without term *b*
Background with term *c*	Background without term *d*

The probability mass function (PMF) for the hypergeometric distribution can be defined as follows:

$$H(a,b,c,d) = \frac{(a+b)!\,(c+d)!\,(a+c)!\,(b+d)!}{(a+b+c+d)!\,a!\,b!\,c!\,d!}$$

For significance level 0.05, the p-value is calculated, applying the Bonferroni correction for *N* significance tests:

$$p_{0.05} = \sum_{x=0}^{x=N} (H(x, b-x, c-x, d+x) \leq 0.05) * N$$

The direction and intensity of verbal enrichment was quantified by the odds ratio:

$$\frac{outcomes\ in\ subset / outcomes\ in\ background}{remainder\ in\ subset / remainder\ in\ background}$$

All comparisons between discrete datasets were performed by forming weighted ranked lists. Initial merging between ranks was unweighted, followed by supervised training of the algorithm to reweight gene candidates by annotations from the literature and online datasets.

2.14: Dataset archiving and localized access of the PubMed citation database

PubMed citations were downloaded in bulk XML format from (ftp://ftp.ncbi.nlm.nih.gov/pubmed/baseline) using cURL and left in compressed form on the hard disk. Citations were accessed by decompressing each archive into memory through a stream using the C compression library zlib (https://zlib.net) and reading the bytes into an XML tree. Accessing citations through this on-demand decompression method ensured that the entire baseline held a manageable memory footprint on the hard drive. Citations were then selected from random indices between 0 and 29100000 and accessed by opening the corresponding archive using the zlib API and surfing the decompression stream to find the PubMed citation with that index.

Ground truth gene context citations were selected at random locations in the PubMed index based on one or more HUGO gene symbols and their matching full names appearing side-by-side in the abstract. The opposite non-gene context citations were selected based on their lack of any string sequence matching a HUGO gene symbol, alias, full name, or former name in the abstract. These were further filtered based on the presence or absence of key words signifying genetic context, such as “gene”, “expression”, “levels”, “sequencing”, and “promoter.” 30000 abstracts of each class were accumulated into a database of 60000 automatically-labelled pairs and stored as a

tab-delimited file (*.tsv), included in Supplementary File S4. The neural recognition gates for genetic and therapeutic context recognition were set at 0.5 and 0.8, respectively.

Therapeutic context citations were sampled at random from the Pubmed citation index based on the appearance of one or more chemicals from the Therapeutic Target Database (TTD) appearing in the body of the abstract, while non-drug context citations were selected based on the absence of these terms. These were also filtered based on the presence or absence of therapeutic context words such as “drug”, “administered”, “pill”, “injected”, and “trial.” 30000 examples of each class were accumulated into an automatically-labelled database of 60000 pairs, which were stored as a tab-delimited file (*.tsv), included in Supplementary File S5.

2.15: Hardware and software specifications

For all experiments performed in this work, a single Asus® Zenbook™ laptop was used with Windows™ 10 (64-bit), an Intel® Core™ i7-7700HQ quad-core CPU clocked at 2.80 GHz processor, and 16 GB of DDR3 RAM. All experiments were parallelized on the CPU, capable of maximally running 8 threads. Since networks were trained using a single worker thread each, up to 8 network training permutations could be parallelized effectively. An Nvidia® GeForce™ GTX™ 1050 Ti was installed on the laptop, but not employed for any of the worker threads. The decision to not port the engine over to the GPU was primarily made to maintain cross-platform compatibility, for simpler troubleshooting, and to ensure that all algorithms were optimized on the CPU first before moving to more powerful hardware.

Engine development was performed using the C++ programming language, compiled using MinGW 7.3.0, in the Code::Blocks 16.01 IDE. The AiDA user interface was created using CVision (https://github.com/DamianTran/cvision), an open source library built upon SFML 3.5.1. Data visualization was performed using R Studio 1.1383 and R version 3.5.1, additional data extraction and manipulation was performed using the dplyr and Bioconductor extensions. Version control was managed by Github via the MSys2 Bash interface (open source repositories at https://github.com/DamianTran). Plots were generated in R using the ggplot2, plotrix, lemon, reshape2, and plotROC libraries.

Chapter 3: Results

3.1: A range-normalized sigmoid activation function achieves high accuracy in benchmark prediction tasks

In order to validate the efficacy of several custom neural network algorithms, neural network models of varying complexity were created to be tested on benchmark datasets of increasing difficulty: Digit-MNIST (Deng, 2012), Fashion-MNIST (Xiao *et al.*, 2017), and CIFAR-10. An extensive list of hyperparameter values at initialization, and validation results for all models are included in Appendix 2. MLP and convolutional configurations were tested to validate the positive effect of the custom convolution algorithm on network convergence. A simple MLP model of layers [784, 128, 10] with range sigmoid activation achieved a maximum 97.18% validation accuracy on Digit-MNIST after 20 epochs of training, which was easily beaten by the convolutional counterpart in Figure 11, which reached 98.64% accuracy with the same amount of training. Dropout was applied to the hidden layer with a dropout probability of 0.15. ReLU activation was applied to the convolutional feature maps with $\alpha = 0.15$.

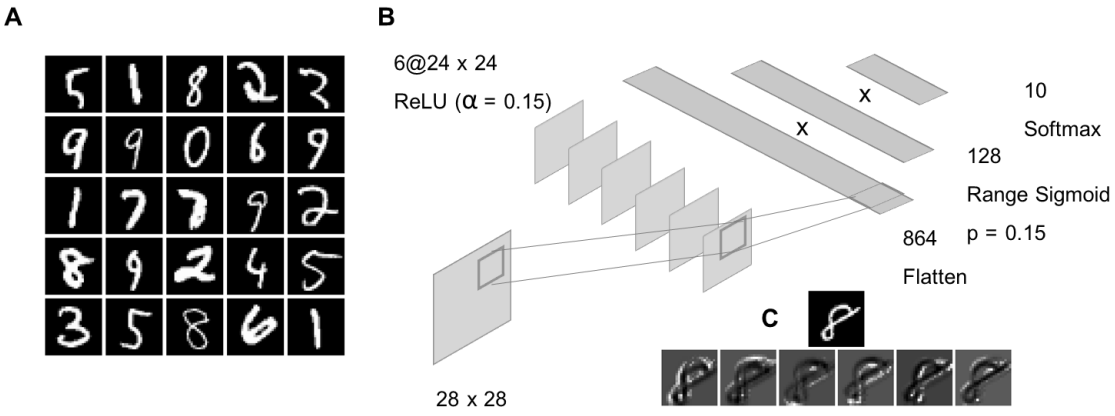


Figure 11. Convolutional architecture applied to numeric digit recognition in Digit-MNIST. (A): A random sampling of 25 Digit-MNIST images, each 28x28 in size. (B): Schematic of a simple convolutional neural network capable of reaching 98.64% validation accuracy after 20 epochs. (C): A feature map sampling of the input image (top) and six convolutional feature maps (bottom) of the neural network while viewing an example of the digit "8."

A similar architecture was created to classify fashion items in Fashion-MNIST, a relatively more difficult benchmark than Digit-MNIST (Figure 12). A simple MLP of depth [784, 128, 10] achieved a maximum classification accuracy of 89.75% on the Fashion-MNIST validation set, but the addition of convolutions enhanced the prediction accuracy to 90.3%. Dropout was applied to the hidden layer with dropout p = 0.1. The range sigmoid activation was used for both hidden layers, and ReLU activation with α = 0.1 was applied to the convolutional feature maps.

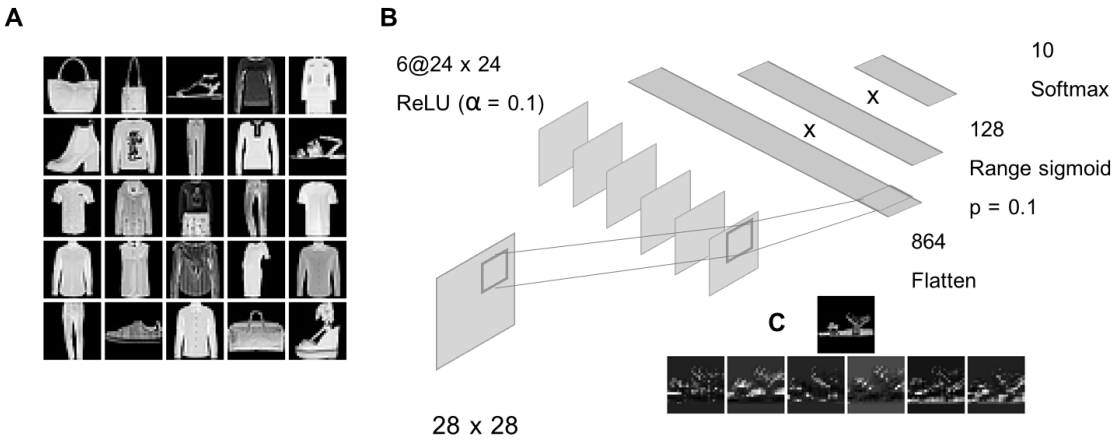


Figure 12. Convolutional architecture applied to low-resolution fashion item recognition in Fashion-MNIST. (A): A random sampling of 25 greyscale images sampled from Fashion-MNIST. (B): A schematic of the neural architecture used to achieve 90.3% validation accuracy after 20 epochs. (C): A feature map sampling of an input image (top) and the 6 convolutional feature maps (bottom) while viewing an example of a "sandal" image.

Finally, a deeper convolutional architecture was created to test the scalability of the engine with regards to pooled convolution layering. A network with 4 convolutional layers (kernels of 3x3, 3x3, 5x5, and 3x3) and an intermittent 2x2 pooling layer connected to an expanded MLP component achieved 62.49% accuracy on the more complex CIFAR-10 dataset (Figure 13).

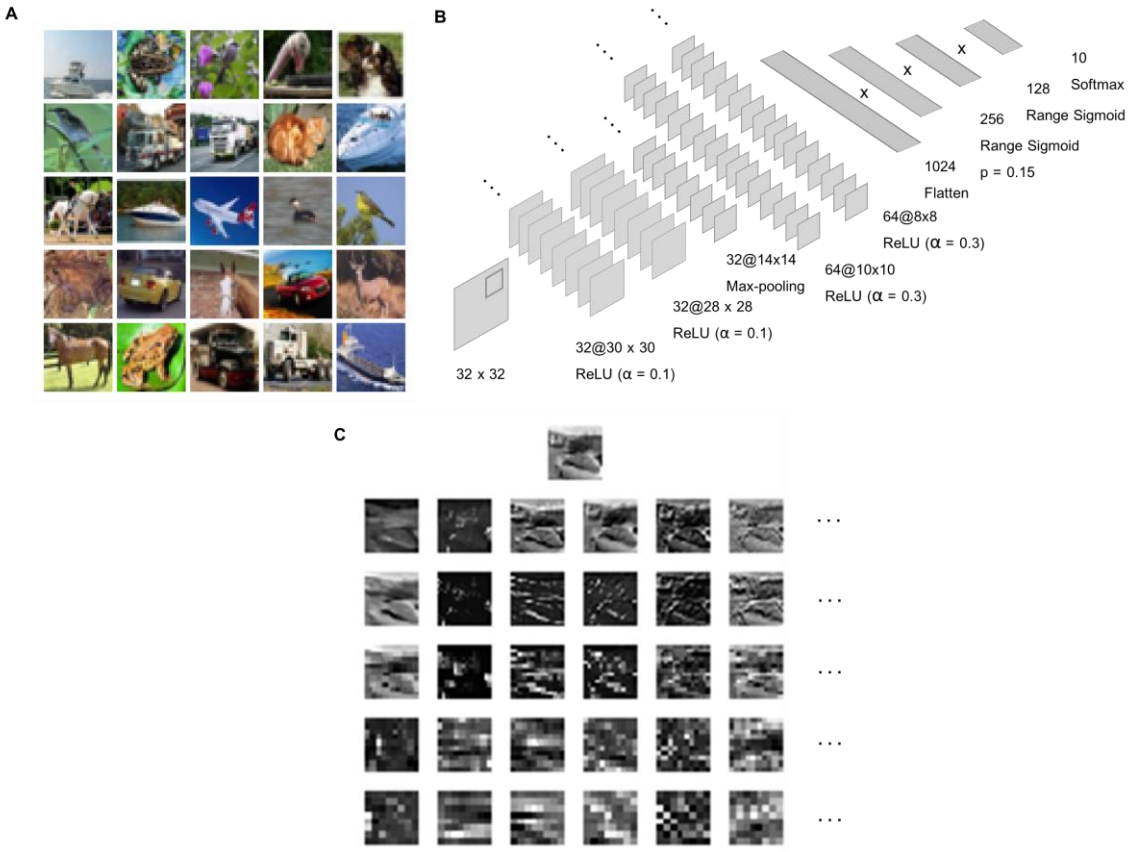


Figure 13. Convolutional architecture applied to low-resolution images in the CIFAR-10 dataset. (A): Random sampling of 25 images from the CIFAR-10 small images dataset. (B): The deep convolutional neural network architecture used to achieve 62.49% prediction accuracy after 20 epochs. (C): A sampling of feature maps at each level of convolution and pooling, from most proximal to the input (top) to most distal (bottom). Only six feature maps were sampled per layer for clarity.

3.2: A text-to-function (T2F) system achieves high prediction accuracy from minimal sparse data

T2F multilayer perceptron and convolutional network architectures were tested to verify if linear convolution of embedding vector dimensions positively contributed to model test scores. The multilayer perceptron configuration shown in Figure 7 and the convolutional

neural network architecture shown in Figure 8 were both tested over 25 randomly permutated trials. The T2F systems were tasked to predict the correct function response to 1265 manually assembled text transcripts. The dataset was split at a 73%/27% training/test ratio, maintaining the ratio across label groups while the random segregation was performed. This resulted in a training epoch size of 928, and a test sample size of 337.

Over the range of random permutations, the convolutional network converged slightly faster (Figure 14a, bottom) and reached a lower loss minimum of 0.1707 (mean prediction confidence = 84.3%) at epoch 59, while the MLP required 72 epochs to reach a loss minimum of 0.2 (Figure 14a, top; mean prediction confidence = 81.9%). Both classes were able to fit to a median of 99.68% of the training data (925/928). The differences in validation performances are exemplified in Figure 14b, where convolutional models collectively scored a median of 83.09% correct on the randomly segregated test set (278/337), 1.19% higher than the median score for the MLP models (81.9%; 276/337). Convolutional models were exhibiting higher generalizability, overfitting 1.08% less than their MLP counterparts. Overall, the overfit margin was large for both models at a median overfit of 16.8% for the convolutional class and 17.89% for the MLP class.

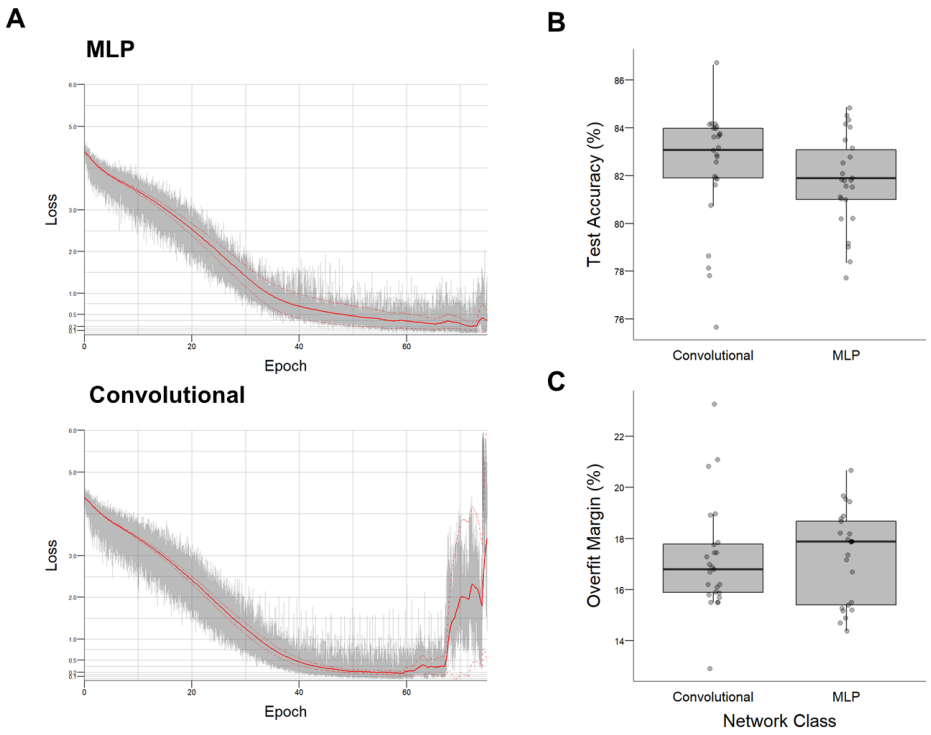
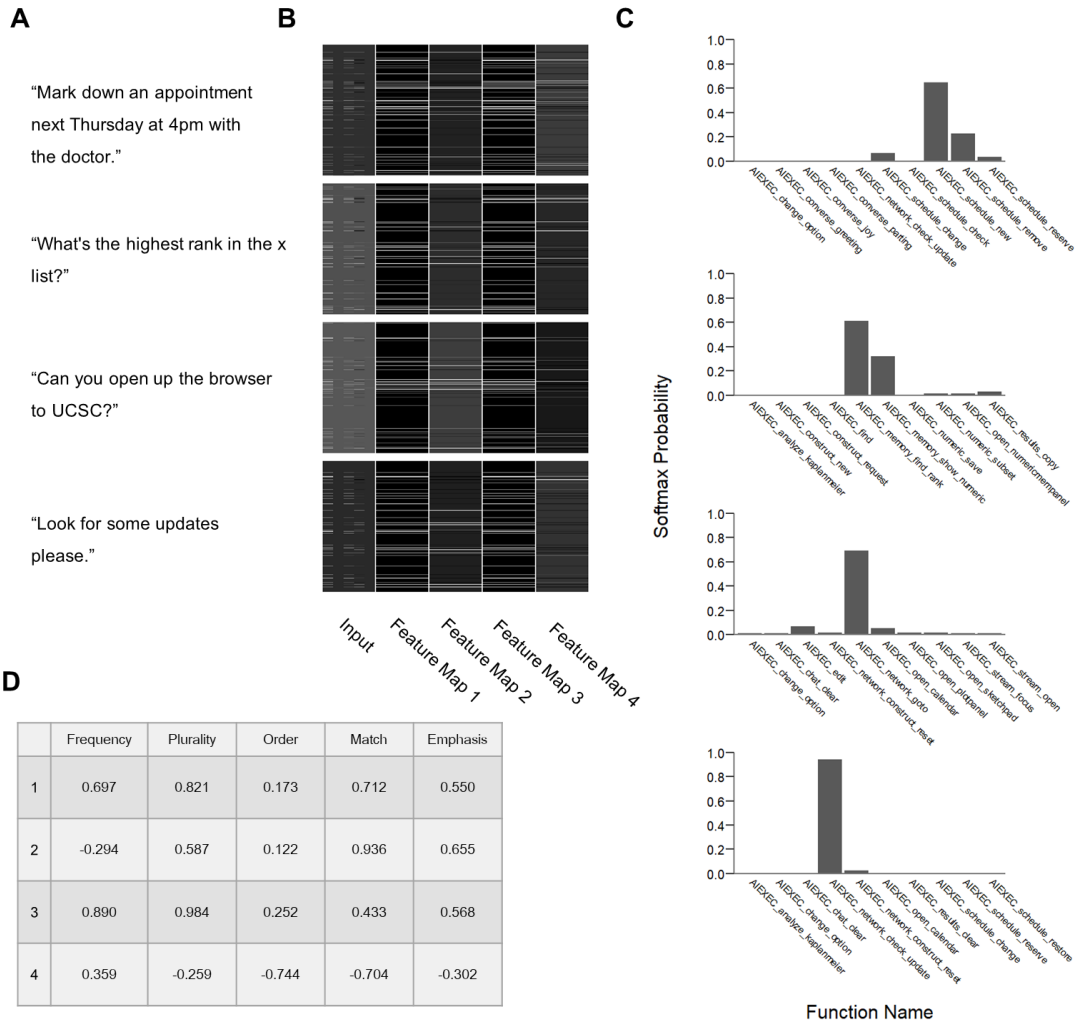


Figure 14. Comparison of collective training results over 25 permutations of T2F data training for MLP and CNN models. (A): Loss progression graphs illustrating both the average of stochastic loss for all 25 permutations (grey) and the average running loss (red) and associated 95% confidence intervals (dotted red) for the multilayer perceptron class (top) and convolutional class (bottom). (B): Staggered box plot comparing test scores on randomly segregated transcript-function pairs over 25 permutations for convolutional and MLP classes. (C): Staggered box plot comparing the overfit margin (inversely proportional to generalizability) for convolutional and MLP classes.

Analysis of the convolutional architecture, which was chosen for future examination due to its marginally better performance, revealed that function classes containing similar transcripts were assigned similar probability scores (Figure 15c). A series of example transcripts were manually procured for demonstration purposes, none of which belong to the original 1265 transcripts in the training dataset (Figure 15a). The first example,

containing words related to scheduling, caused the network to produce top 4 probabilities centered on 4/6 labelled scheduling functions (schedule_new: 64.6%, schedule_remove: 22.6%, schedule_change: 6.7%, schedule_reserve: 3.2%, all others: < 2.9%). The second example, containing words related to the access and manipulation of ranked list constructs, caused the model to predict outcomes centered around ranked list-related functions (memory_find_rank: 61.2%, memory_show_numeric: 31.8%, results_copy: 2.65%, numeric_subset: 1.57%, open_numericmempanel: 1.29%, all others: < 1.35%). The linear convolutional filters (feature maps shown in Figure 15b) each learned different polynomial characteristics of the embedding dimensions (Figure 15d).



D

	Frequency	Plurality	Order	Match	Emphasis
1	0.697	0.821	0.173	0.712	0.550
2	-0.294	0.587	0.122	0.936	0.655
3	0.890	0.984	0.252	0.433	0.568
4	0.359	-0.259	-0.744	-0.704	-0.302

Figure 15. Convolutional kernel analysis and functional response predictions for text transcript examples. (A): Manually-curated text transcripts not belonging to the original 1265 transcripts of the T2F text database. Examples are matched row-wise to panels in (B) and (C). (B): Input images and convolutional feature maps for the best-performing convolutional T2F network while reading each of the transcript examples in (A). Images have been transformed for visibility; true input map dimensions were 5 x 317 and true feature map dimensions were 1 x 317. (C): Bar plots of softmax prediction probabilities for the top 10 predictions made for each transcript in (A). All probabilities sum to 1.0. (D): Convolutional

kernel weights for the best performing convolutional T2F model for each filter (rows) and dimension (columns).

3.3: Discovery of gene target candidates with functional genomic screening potential in acute myeloid leukemia (AML)

A genome-wide ranked list analysis comparing overall survival predictions and disease-free survival predictions was performed between patients exhibiting above- and below-median gene expression for every gene with transcript expression data in LAML-TCGA (Figure 16a). The ranked list analysis was proficient at enriching for genes with enhanced expression in functionally-validated leukemic stem cell (LSC)-containing, sorted peripheral blood fractions based on analysis of a cohort of 78 AML patients (Figure 16b). Stratification of patients by high-ranking prognostic genes predicted significantly poorer survival outcomes for above-median expression cohorts by up to 40% over 5 years (Figure 16c).

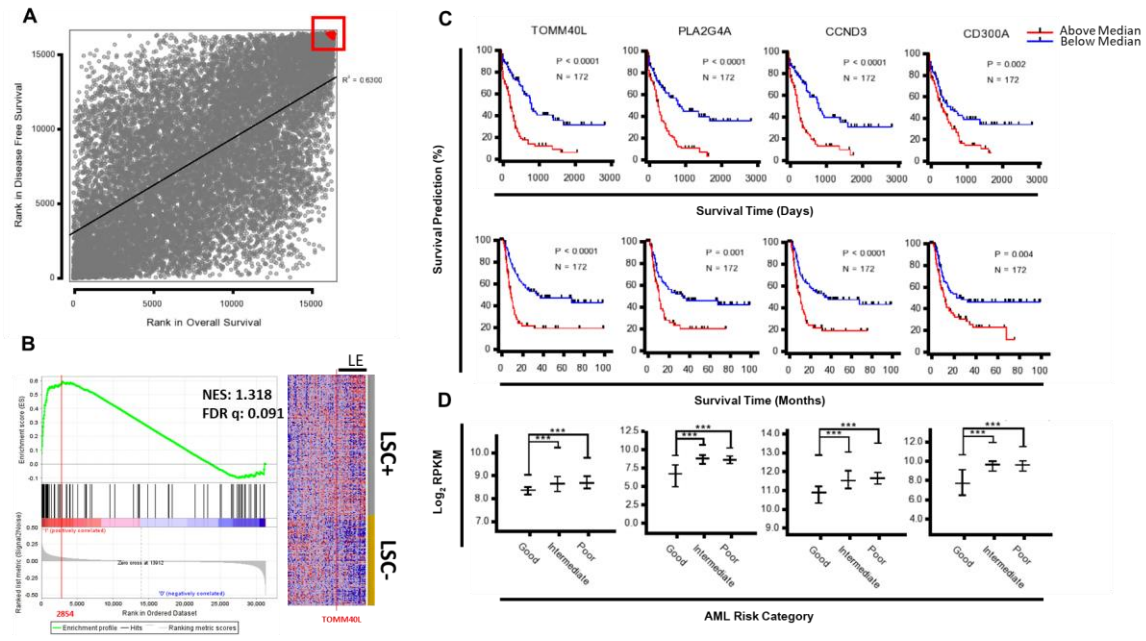


Figure 16. Weighted ranked list algorithms establish a metric for genome-wide prognosis. (A): Scatter plot of all genes with expression data in LAML-TCGA ranked by their ability to stratify patients by overall survival vs. disease-free survival based on above- and below-median expression (Spearman R^2 = 0.62). Individual scores were calculated by the Kaplan-Meier method to correlate higher gene expression with poorer patient outcomes. A selection of the top 100-ranked genes was made (red) to assess at a higher resolution. (B): Gene set enrichment analysis plot and clustered expression heatmap demonstrating positive enrichment of the top 100 prognostic genes in LSC+ samples from a dataset of 227 sorted peripheral AML blood fractions (GSE76009). The leading edge (LE) is indicated by vertical red lines (index 2854). (C): Kaplan-meier survival plots visualizing factors of overall survival (top row) and disease-free survival (middle row) for 4 selected genes among the 100 most prognostic ranks. Patient cohorts were stratified from LAML-TCGA (n = 173) based on above- (red) and below-median (blue) expression of the indicated genes (above). P-values were calculated by the log-rank method. (D): Quantile plots of gene expression (log_2-normalized RPKM) including the bottom quartile, median, and top quartile of patients in LAML-TCGA grouped by their cytogenetic risk category (*** = $p < 0.001$).

Following prognostic gene prediction, multiple additional factors of safety were considered, notably low expression across the normal hematopoietic hierarchy, as well as lowest expression in primitive sorted blood fractions. Iterative ranked-list analyses were performed on a cumulative base of 520 samples from LAML-TCGA (n = 173), GSE42519 (n = 34), GSE76009 (n = 227), and GSE12417 (n=86). Ranked gene lists of length 12000 – 18000 were generated, merged, and re-ranked based on the mean of ranks across datasets (dataset weights were initialized evenly). The complete list contained 11444 gene rank predictions for 6 different factors (4 prognosis, 2 safety; Table S1). Selecting among the top 100 of these predictions and mapping back to the predicted prognosis list revealed a high degree of overlap with the upper proportion of prognostic gene predictions (Figure 17a). The enrichment of these genes in LSC-containing AML blood fractions dropped relative to the high positive enrichment of the binary list analysis, while retaining a component of positive enrichment that could be further investigated (Figure 17b). Analyses of overall survival across studies of both treated (LAML-TCGA) and untreated (GSE12417) AML patients showed the above-median expression of these genes was associated with poorer overall survival outcomes by up to 30% over 5 years (Figure 17c).

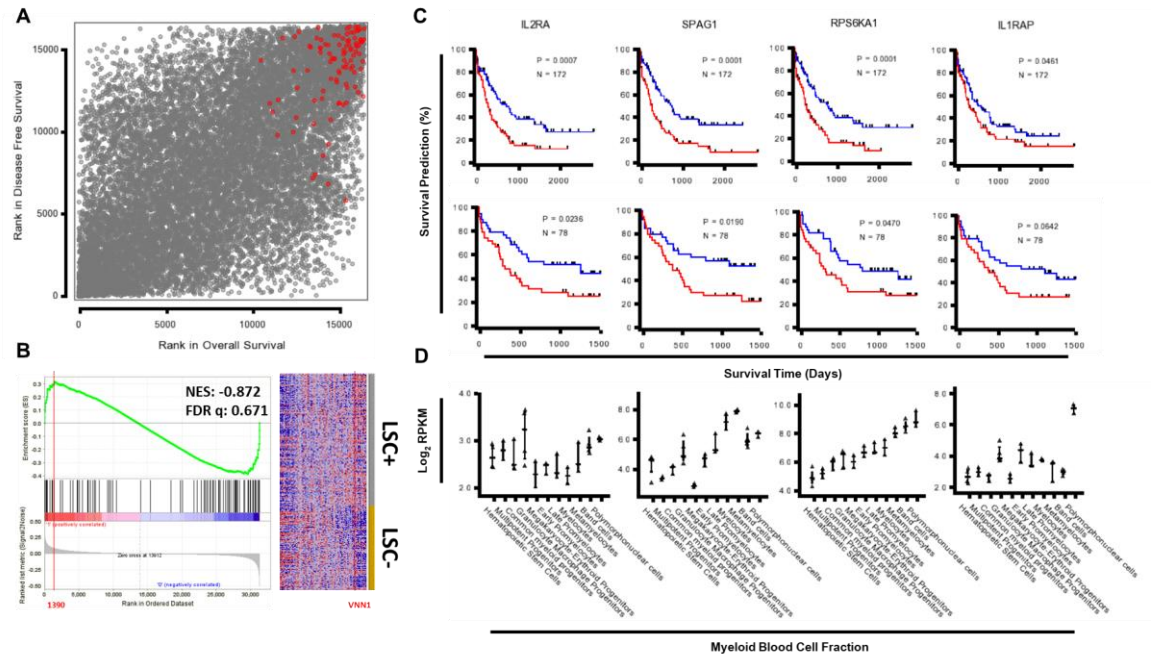


Figure 17. Weighted ranked list algorithms discover gene target candidates matching several required clinical characteristics. (A): Genome-wide prognosis scatterplot created as in figure 2a, with a selection of the top 100-ranked multi-factor genes (red). (B): Gene set enrichment analysis plot and clustered expression heatmap revealing a subset of genes driving positive enrichment in LSC+ samples from a dataset of 227 sorted peripheral AML blood fractions (GSE76009). The positive leading edge (LE) is indicated by vertical red lines (index 1390). (C): Kaplan-meier survival plots for 4 selected genes among the top 100 multi-factor ranks. Patient cohorts were stratified from LAML-TCGA (top row, n = 173) and GSE12417 (middle row, n = 86) based on above- (red) and below-median (blue) expression of the indicated genes (above). P-values were calculated by the log-rank method. (D): Quantile plots of gene expression (log_2-normalized RPKM) including the bottom quartile, median, and top quartile of myeloid blood samples in GSE42519 grouped by cell type. Cell types were ordered left-to-right by increasing myeloid lineage.

In the analysis of these ranked lists, gene candidates *IL1RAP*, *RPS6KA1*, *IL2RA*, and *SPAG1* were selected as optimally-scoring variables matching all 6 factors of

consideration. From among the most prognostic gene predictions, *CALCRL* (Calcitonin receptor-like receptor), *CCND3* (Cyclin D3), *FAM124B* (Family with sequence similarity 124 member B), and *FHL1* (Four and a half LIM domains 1) were determined to be the most positively associated with poor survival outcomes in all AML studies investigated (RNAseq V2 RSEM (n = 173) and Affymetrix U133A (n = 173) microarray profiling of LAML-TCGA, Affymetrix Human Genome U133plus profiling of untreated AML in GSE12417).

CALCRL codes for a G-protein coupled receptor (GPCR) that interacts strongly with the receptor activity modifying protein (RAMP) family of type I transmembrane proteins (specifically, *RAMP1, RAMP2,* and *RAMP3*) required for the transport of calcitonin-receptor-like-receptor (*CLCR*) to the plasma membrane (Archbold *et al.*, 2011; Dackor *et al.*, 2007). *CCND3* encodes for a cyclin protein that is reported to be upregulated in several leukemias, as well as other non-hematopoietic malignancies (Büschges *et al.*, 1999; Liu *et al.*, 2015; Smith *et al.*, 2005). *FAM124B* encodes a mainly nuclear-localized protein speculated to be involved in CHARGE syndrome due to evidence showing it to be a direct binding partner of chromodomain helicase DNA binding protein 8 (CHD8); CHD8 forms a complex with CHD7, a protein known to be mutated in CHARGE syndrome (Batsukh *et al.*, 2012). *FHL1* encodes a member of the four-and-a-half-LIM-only family of proteins that are characterized by two highly conserved, tandemly arranged zinc-finger domains each with four highly conserved cysteines binding a zinc atom(Zipfel and Skerka, 1999). It is expressed in cell-type-specific ways, notably in skeletal muscle (Morgan and Madgwick, 1999), and mutant variants have been found to

be associated with a variety of human myopathies (Chen *et al.*, 2010; Schessl *et al.*, 2008; Windpassinger *et al.*, 2008).

3.4: Expression of highly ranked candidates is positively associated with the presence of AML mutational hotspots

Mutational enrichments in the high ranks of the consolidated ranked list output were investigated by performing verbal enrichment analysis on annotation datasets. Data bridges were formed between LAML-TCGA and the associated cBIOPortal dataset containing mutation events for all 173 AML patients. This allowed for the automatic discovery of mutation enrichments associated with patient subsets selected by expression levels of *IL1RAP* (Interleukin 1 receptor accessory protein), *RPS6KA1* (Ribosomal protein S6 kinase A1), *IL2RA* (Interleukin 2 receptor subunit alpha), and *SPAG1* (Sperm associated antigen 1) (Figure 18a; data included in Supplementary File S6). Analysis of verbal enrichments for each individual gene in above-median expression cohorts (n = 86) for *IL1RAP*, *IL2RA*, and *SPAG1* revealed positive correlations to *FLT3* (Fms related tyrosine kinase 3) mutation frequencies. The *FLT3* gene encodes a tyrosine kinase protein that is a cell surface receptor for the FLT3LG cytokine (Shurin *et al.*, 1998). Activating mutations such as internal tandem duplications (ITDs) in the *FLT3* locus are well-known to be associated with AML and are present in about a third of all AML cases (Cortes *et al.*, 2016; Kindler *et al.*, 2010; Kottaridis *et al.*, 2001; Meshinchi *et al.*, 2006; Zarrinkar *et al.*, 2009).

IL1RAP encodes a coreceptor of IL1R1 (Interleukin 1 receptor type 1) in the interleukin 1 receptor complex, which initiates signalling events resulting in the activation of interleukin 1-responsive genes (Lingel *et al.*, 2009; Tominaga *et al.*, 2000). It's been recently highlighted as a potential therapeutic target in AML due to its increased presence on the surface of AML stem cells, and its involvement with several AML signalling pathways (Ågerstam *et al.*, 2015; Askmyr *et al.*, 2013; Mitchell *et al.*, 2018). The protein product of *RPS6KA1* (also known as *P90RSK*) is a serine/threonine-protein kinase that acts downstream of MAPK1/ERK2 and MAPK3/ERK1 signalling (Dalby *et al.*, 1998; Shimamura *et al.*, 2000; Wingate *et al.*, 2006). Little is known about its role in any genetic diseases, but it has been investigated in the context of kidney fibrosis (Lin *et al.*, 2019). *IL2RA* encodes the alpha subunit of the interleukin-2 receptor involved in the regulation of immune tolerance by control of regulatory T cell activity (Bezrodnik *et al.*, 2014; Goudy *et al.*, 2013). mRNA expression of *IL2RA* was very recently reported to be an independent prognostic factor in intermediate risk AML (Du *et al.*, 2019). The protein product of *SPAG1* is not well-studied but is known to bind GTP and have GTPase activity (Lin, 2001). Assumptions have been about its role in the cytoplasmic assembly of ciliary dynein arms, and potentially its participation in the process of fertilization due to the similarity of its sequence to a previously-categorized 75-kD peptide involved in infertility (Zhang *et al.*, 1992).

Investigation of mutation types revealed that insertion mutations were overrepresented among mutation events ($p < 0.0001$, odds ratio > 1.7). *IL2RA* was highly associated with splice region variant labels ($p < 0.05$, odds ratio > 3), while *RPS6KA1* was conversely

very lowly associated with splice activity ($p < 0.01$, odds ratio < 0.5). From the four optimal candidates *IL1RAP* was the most strongly associated with *FLT3* mutation labels ($p < 0.0001$, odds ratio = 2.61). A subset of patients uniquely expressing above-median levels of all four genes simultaneously (n = 19) was selected from the dataset and assessed via data bridge to the mutation dataset. Collectively, the genes were extremely enriched for *FLT3* mutation incidents ($p < 0.0001$, odds ratio = 3.28) and insertion mutation types ($p < 0.0001$, odds ratio = 2.41).

Analysis of *FLT3* gene signatures by GSEA demonstrated that genes predictive of AML characterised by internal tandem duplications (ITD) of *FLT3* (Valk *et al.*, 2004) were highly positively enriched among the high ranks of the consolidated ranked list (Figure 18b; ES = 0.596, $p < 0.0001$, FDR q = 0.049). These genes had been identified by the gene list authors as part of a larger project with the core aim of discovering prognostic gene signatures in AML. Among the leading edge of the enrichment results, *IL2RA* ranks first, with *IL2RAP* behind in second.

Additional analysis revealed above-chance incidences of *NPM1* mutation events for patient cohorts selected by above-median expression of *IL1RAP* (N = 86, p = 0.007, odds ratio = 1.69). Significant enrichment for *NPM1* mutation events was not observed for the other selected candidates, but patient groups selected based on above-median expression of all candidates simultaneously exhibited a slight enrichment for these events (N = 19, p = 0.039, odds ratio = 1.88). Gene set enrichment analysis for a gene list previously reported to be upregulated in *NPM1*-mutant AML (Verhaak, 2005) showed strong

enrichment at the top ranks of the consolidated list (Figure 18c; ES = 0.596, p < 0.0001, FDRq = 0.049, LE = 75/131).

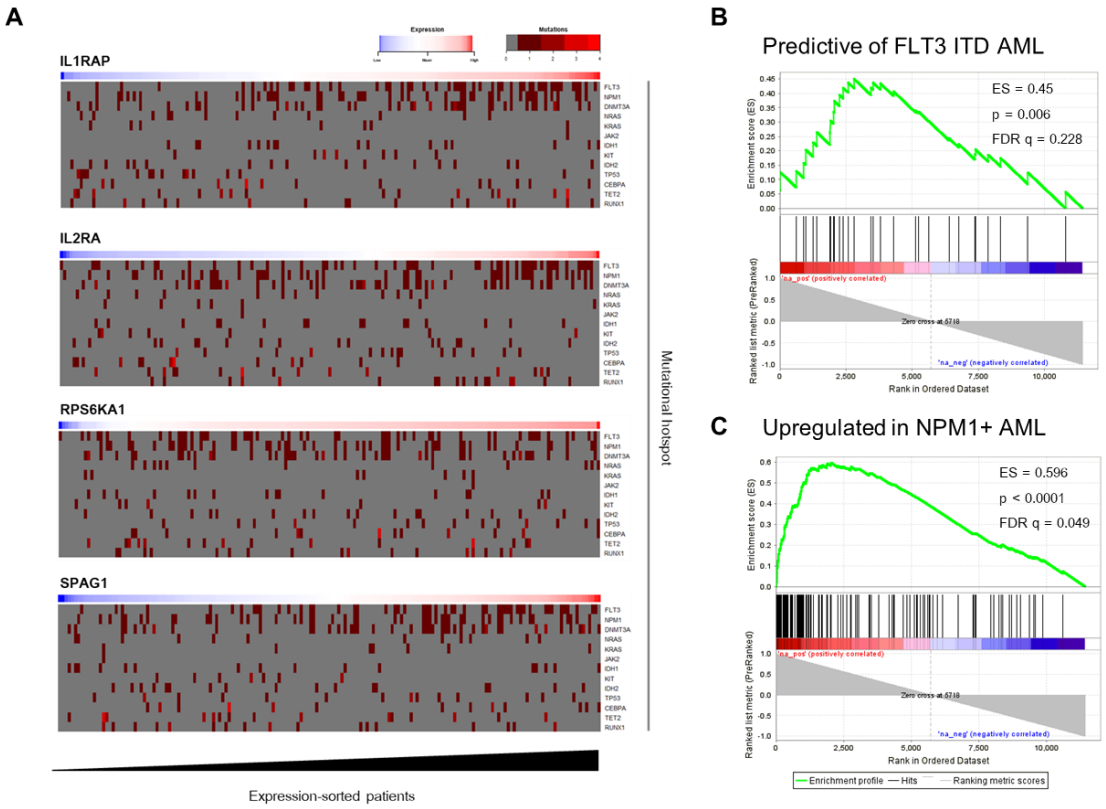


Figure 18. AML mutational hotspots correlate in frequency with candidate gene expression. (A): Heatmaps of mutation events in 173 patients in LAML-TCGA are sorted in increasing order of the expression of 4 high-interest gene outputs. Mutation events are color coded based on the copy number effect of each event. (B): GSEA enrichment plot for a list of genes reported by Valk *et al.* (2004) to be predictive of *FLT3* ITD AML. 30/40 genes could be mapped to the ranked list output, 15/30 comprised the leading edge. (C): GSEA enrichment plot for a list of genes reported by Verhaak et al (2005) to positively correlate in expression with NPM1 mutation events. 131/183 gene list members could be mapped to the ranked list, 75/131 were found in the leading edge.

3.5: Context-recognition CNNs identify therapeutic contexts in the biomedical literature

A genetic and therapeutic context recognition models were trained on databases of 60000 randomly selected, filtered citation abstracts that were randomly segregated into 42000 training examples and 18000 test examples (Supplementary Files S4 and S5). The genetic context detection model achieved a validation accuracy of 95.73% after 20 epochs, with a receiver operating characteristics (ROC) area under the curve (AUC) of > 0.99 (Figure 19a). The therapeutic context recognition model achieved a test accuracy of 83.7% after 20 epochs, with a ROC AUC of 0.92 (Figure 19b). Both models exhibited relatively even true positive and true negative rates, while the genetic context recognition model showed a bias toward false positives, and the therapeutic context recognition showed a marginal bias toward false negatives.

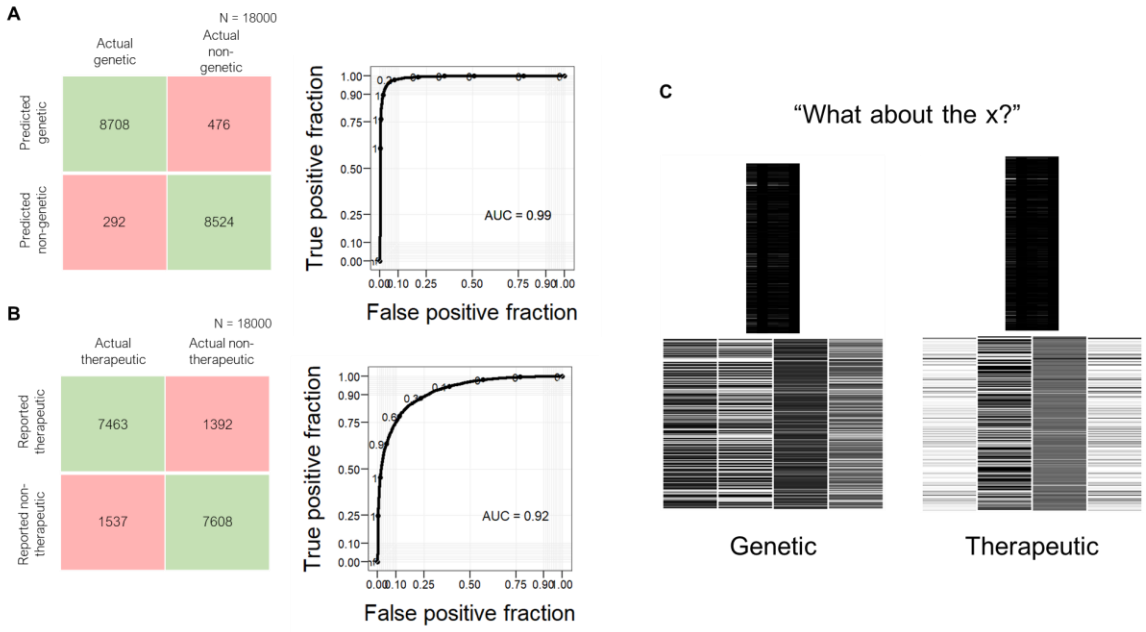


Figure 19. Validation results for context recognition models on PubMed citation abstracts. Genetic (A) and therapeutic (B) context recognition models were trained on 42000 pre-filtered, randomly selected PubMed abstracts for 20 epochs and validated on 18000 pre-filtered, randomly selected PubMed abstracts. (A): Confusion matrix (left) and ROC curve (right) for the best-performing gene context recognition model. Validation score: 95.73%; AUC (area under curve) 0.9916. (B): Confusion matrix (left) and associated ROC curve (right) for the best-performing therapeutic context recognition model. Validation score: 83.72%, AUC: 0.9168. Labels on ROC line points indicate threshold values. (C): Input embedding images (above) and convolutional feature maps (bottom row) for the transcript example "What about the x?" shown for the best-performing genetic (left) and therapeutic (right) recognition models.

Neural network architectures with the highest validation scores were selected to perform context recognition on the cumulative PubMed citation database. Over 29 million citations were downloaded locally in compressed form and scanned to detect articles containing the top and bottom 25 genes of the consolidated ranked list. Citation abstracts that contained matching gene acronyms were turned into 2D embedding matrices and scanned by both the genetic and therapeutic context recognition models. When neural model detection gates were unmodified (detection threshold = 0.5), 216 matching PubMed citations were discovered that both networks signalled positive for genetic and therapeutic contexts. For the bottom 25 genes, 336 matching citations were identified that met the network thresholds. 272 abstracts for these genes did not meet either the threshold for the genetic context model or the therapeutic context model and were filtered out. A summary of these results is displayed in table 3 below (complete list in Supplementary File S7).

Gene	List Rank	Abstract Title	PMID
IL2RA	1	Targeting Pseudomonas exotoxin to hematologic malignancies.	8562907
RPS6KA1	14	Gene expression patterns of hippocampus and cerebral cortex of senescence-accelerated mouse treated with Huang-Lian-Jie-Du decoction.	17805973
VNN3	4	Pharmacologic concentrations of ascorbic acid cause diverse influence on differential expressions of angiogenic chemokine genes in different hepatocellular carcinoma cell lines.	19932582
CYP7B1	13	Effect of ribavirin, levovirin and viramidine on liver toxicological gene expression in rats.	14635270
TCF3	11433	Inhibition of protein-protein interactions: the discovery of druglike beta-catenin inhibitors by combining virtual and biophysical screening.	16568448
CCND2	11440	Gamma-secretase inhibitors reverse glucocorticoid resistance in T cell acute lymphoblastic leukemia.	19098907

UBB	11419	Effects of dimethyl sulphoxide and dexamethasone on mRNA expression of myogenesis- and muscle proteolytic system-related genes in mouse myoblastic C2C12 cells.	18835828
ITM2A	11438	Enhanced ITM2A expression inhibits chondrogenic differentiation of mesenchymal stem cells.	19541402

Table 3. Summary of drug discovery results facilitated by deep text mining. From a list of abstracts enriched for genetic and therapeutic contexts, a handful of promising abstracts were selected for display above.

An abstract discussing a promising pharmacological inhibitor for *IL2RA*-related hematologic malignancies is discussed in PMID 8562907 (Kreitman and Pastan, 1995). Excitingly, the authors of this memo report an immunotoxin targeting the interleukin-2 receptor alpha subunit that underwent clinical trials in patients exhibiting various leukemias in 1995. It is a lowly-cited report (75 citations at the time of writing) that failed to appear via PubMed's online search tool when the queries "IL2RA", "IL2RA therapeutic", and "IL2RA inhibitor" were attempted. Another difficult-to-find entry, published in *Neuroscience Letters* in 2008, was PMID 17805973 which discusses a traditional Chinese medicine therapy called the Huang-Lian-Jie-Du decoction (Zheng *et al.*, 2008). In this paper, with 33 citations at the time of writing, the authors demonstrated using RT-qPCR that they to be able to modulate the expression of

RPS6KA1. The article is a less conventional example of a therapeutic application compared to more conventional pharmacological examples, such as PMID 19932582, a paper discussing pharmacological ascorbic acid and its effect on the expression levels of VNN3, among other targets, at different concentrations. In PMID 14635270, *CYP7B1* was demonstrated to have been inhibited four-fold by Levovirin, an L-enantiomer of Ribavirin, which is a common treatment for chronic hepatis C (Fang *et al.*, 2003). Importantly, its alternative pathway to fulfilling its normal role in cholesterol metabolism via interaction with *CYP27* was not affected. The authors concluded that high doses of Levovirin did not cause significant dysregulation of liver toxicological genes, indicating viability for the use of this compound as a safe therapeutic. Levovirin presents potential for further investigation due to its inhibition of a highly-ranked candidate gene in AML (further discussed in *4.2*).

Studying results procured from the bottom 25 genes of the consolidated ranked list yielded equally fruitful results. PMID 19098907 published in *Nature Medicine* discusses findings that glucocorticoids enhanced the expression of *CCND2*, which the authors suggested could be applied in combination with gamma secretase inhibitors (GSIs) for glucocorticoid-resistant T-cell acute lymphoblastic leukemia (Real *et al.*, 2008). PMID 19541402 published in *Differentiation* contains a protocol for forced induction of IMT2A in mesenchymal stem cells, resulting in preservation of their primitive states (Boeuf *et al.*, 2009).

3.6: AML and LSC signatures are enriched at the extremes of the consolidated ranked list

A GSEA analysis was performed using the total curated list of 3173 gene signatures available through MSigDB (supplemental file S6). 365 signatures were found to be significantly enriched ($p < 0.05$) at the top half of the list, while 227 signatures were significantly enriched at the bottom half of the list. Among the gene signatures most highly enriched at the high ranks of the candidate list, several AML- and LSC-related groups were identified (Figure 20). A list of 40 genes previously reported to hold high prognostic value in M4 and M5 FAB subtypes (Valk *et al.*, 2004) mapped to 22 genes on the ranked list, all of which were enriched in the top half of the rankings (NES = 0.78, $p < 0.0001$, FDR $q < 0.0001$, LE = 20/22). Overall the Valk *et al.* study used prediction analysis of microarrays (PAM) and nearest shrunken centroid analysis to predict poor survival and event-free outcomes for patients with a combination of aberrations in these genes belonging to cluster 5 (Valk *et al.*, 2010). 4 of these genes were leukocyte immunoglobulin-like receptor family members (*LILRB1*, *LILRA1*, *LILRA6*, and *LILRB3*; mean rank = 294.25). Another highly-enriched gene signature contained 131/183 genes previously reported to be upregulated in NPM-positive AML by Verhaak at al. in 2005 (NES = 0.597, $p < 0.0001$, FDR $q < 0.049$, LE = 75/131). This leading edge also contained many leukocyte immunoglobulin-like receptor members such as *LILRB1*, *LILRA3*, *LILRA1*, and *LILRA6* (mean rank = 199.5). A list of genes found by Gentles *et al.* (2010) to be upregulated in AML LSCs over corresponding progenitor fractions was also enriched at the high ranks of the candidate list (NES = 0.516, $p = 0.004$, FDR q =

0.149, LE = 12/24). These genes were obtained from a retrospective study of AML tumor microarray data (n = 1047; Gentles et al, 2010).

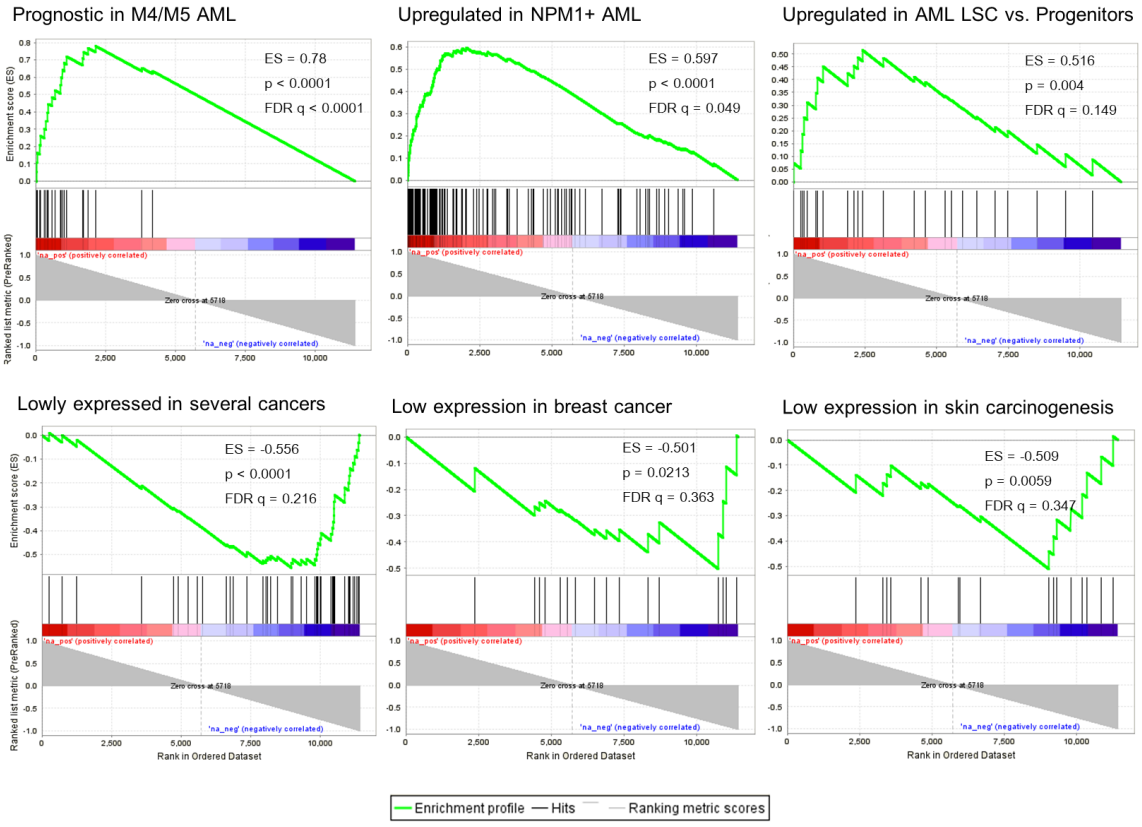


Figure 20. Gene set enrichment analysis results for high- and low-ranked gene lists. (Top): Enrichment results for curated lists with high enrichment scores at the high ranks of the consolidated list. Results chosen for visualization were selected from among the 119 gene sets significantly enriched at FDR < 25%. (Bottom): Enrichment results for curated lists with low enrichment scores at the low ranks of the consolidated list. Results chosen from visualization were selected from among the 92 gene sets significantly enriched at nominal p-values below 1%.

On the bottom half of the gene list, a gene signature reported to contain members expressed at low levels in nasopharyngeal, breast, and liver tumors (Liu *et al.*, 2008) was

found to be highly enriched for 46/79 of those members (NES = -0.556, $p < 0.0001$, FDR $q = 0.216$, LE = 26/46). Another gene signature found by Naderi at al. (2007) found to contribute to poor survival outcomes when lowly expressed was found to be highly enriched at the bottom half of the candidate ranked list (NES = -0.501, $p = 0.213$, FDR $q = 0.363$, LE = 6/16). 16 from the 18 genes reported in this work could be mapped to the consolidated candidate list. Lastly, a group of genes reported by Schlingemann *et al.* (2003) to have been downregulated during carcinogen-induced oncogenic transformation in mouse models was found to be enriched at the bottom half of the candidate list (NES = -0.509, $p = 0.0059$, FDR $q = 0.347$, LE = 7/17).

Chapter 4: Discussion

4.1: Summary

The task of narrowing down potential avenues of study in research is normally an arduous process, performed manually by extracting and visualizing small sections of experimental data using a variety of proprietary software options, addition to perusing the published literature by using internet search tools for each candidate. To facilitate this process, an automated platform, called AiDA, was created with the intention of becoming an end-to-end solution covering the entire workflow from raw data analytics to systematic literature review. This application will be powered by the T2F (text-to-function) neural technology, which proved to be able to learn organically from a large database of paired text-function examples. The chatbot interface was developed using the CVision open source library, which is provided to allow the user to ask more free-form questions that would be part of natural thought processes during lead generation.

Using the T2F system connected to a series of ranked list consolidation algorithms, a concise group of candidate genes were identified for further investigation in the acute myeloid leukemia context. The relatively condensed list of candidates, stratified by their ranks on a larger consolidated list, was manageable enough to be more thoroughly assessed using conventional bioinformatics tools. The power of automated literature searches enabled rapid and concise therapeutics discovery on a scale not previously reported. These tools offer transparency in the way they operate, as shown in the

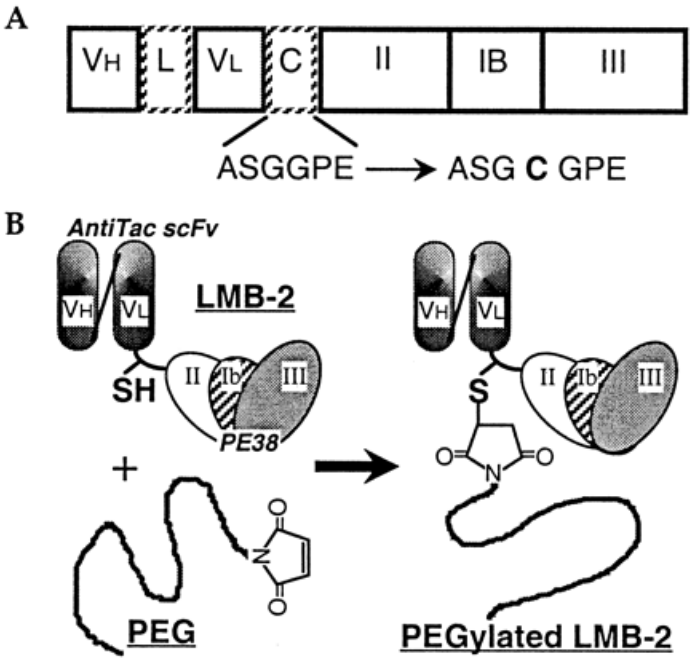
multiple graphs and charts provided as evidence for empirical deep learning throughout this thesis.

4.2: Critical analysis of therapeutic discoveries

Use of the two-tier context recognition system comprising of a genetic context neural gate and a therapeutic context neural gate showed that a vast body of literature could be mined and filtered to present the investigator with a select few promising leads. The lower accuracy of the therapeutic recognition gate (83.7%) compared to the genetic recognition gate (95.7%) could be offset by increasing the threshold value to 0.8/1.0. In this way, only abstracts that very heavily activated the gate would be flagged as literature describing a therapeutic application. This tuning result shows that, in downstream applications, the logic gate thresholds for both context recognizers may be experimented with for different effects on sensitivity and specificity. The requirements for each investigator would likely be different, but it's likely that a conservative search with high gate thresholds may be initiated first to find fast results. Should results fail to appear, the gate thresholds may be lowered to display more results, until eventually all unfiltered results matching the raw query are presented.

From *3.5*, several promising therapeutic compounds were identified with recyclable potential in AML. To highlight the background and progress of a select example, the pseudomonas exotoxin derivative, Anti-Tac(Fv)-PE38 (also known as immunotoxin LMB-2), was mentioned in a report by Kreitman and Pastan (1995) to have entered clinical trials for *IL2RA*-positive leukemia, lymphoma, and Hodgkin's disease. It was reported in several papers a few years later to have some non-specific cytotoxicity (Onda

et al., 1999, 2000; Tsutsumi *et al.*, 2000), but overall demonstrated promising results in phase I clinical trials (Kreitman *et al.*, 2000). As of yet, the immunotoxin has not been investigated in applications specific to AML.



(Tsutsumi *et al.*, 2000)

The above figure, taken from a paper by Tsutsumi *et al.* (2000) discusses methods for more site-specific administration of the immunotoxin, and applies polyethylene glycol (PEG) as a delivery vessel. Kreitman and Pastan had reported that a single patient had developed cardiomyopathy during the course of phase I trials—perhaps for similar reasons the immunotoxin on its own hasn't been overtly celebrated. Despite this, clinical trials have been progressing, with the most recent reports about the compound being published in 2009 (Kreitman, 2009) and 2012 (Singh *et al.*, 2012). Kreitman had

reported that the immunotoxins were most successful after the failure of standard chemotherapy (Kreitman, 2009).

The context recognition system discovered Levovirin reported in PMID 14635270, an inhibitor targeting *CYP7B1*, which is a gene candidate ranking 13th on the consolidated ranked list produced using the AiDA platform. Levovirin had been previously reported as the L-enantiomer of Ribavirin, a common treatment for chronic hepatitis C (Fang *et al.*, 2003). While *CYP7B1* is expressed mainly in the liver, it was reported to have been influenced unexpectedly by a Fanconi anaemia-associated gene, *FANCC* (Fanconi anaemia complementation group C) (Zanier *et al.*, 2004). The Fanconi anaemia (FA) condition is a hematological disorder that is defined by an aplastic anaemia, bone marrow failure, and pancytopenia (Joenje and Patel, 2001). Many reports link FA and its related genetic pathways to enhanced likelihood of leukemia and potential contributions to leukemogenesis (Alter, 2014; Auerbach and Allen, 1991; Du *et al.*, 2016; Rosenberg *et al.*, 2003). For this reason, and for the ranked list evidence supplied toward *CYP7B1* being a therapeutic target in AML, it could be recommended to investigate the gene's involvement in AML and the potential therapeutic value of the Levovirin, and its enantiomer Ribavirin. At the time of writing, *CYP7B1* has not been explored as a specific target of therapeutics against AML.

4.3: T2F challenges and future directions

What the deep learning validation results ultimately showed was that the T2F system, and its context recognition derivative, were able to learn from text databases of varying size and sparsity and produce generalizable models that make sensible predictions about

unstructured text. These, while not perfect, proved in the T2F case that many potential unseen user requests could be adequately responded to. In the context recognition case, the models demonstrated that the vast embodiment of accumulated literature could be narrowed down to a manageable number of high-priority tractable leads. The deep learning literature selection method requires refinement to bring the accuracy of the model up to par with the rigor of systematic review, but this work has shown that it may be possible, with enough layering of context recognition, that the process can at least be expediated by deep learning methods.

The present learning model is adequate to support the claim that several human qualities such as unstructured text interpretation and context recognition can be partially automated and applied iteratively to analyze a large body of evidence. It can be further improved by expanding the training sets to encompass greater representation in the total PubMed citation baseline. Further metrics such as the false discovery rate would be desirable to supplement the reported "true positives," however any permutative methods could take exorbitant amounts of time and hardware resources to demonstrate significant FDRs together with each independent search. A new method will have to be implemented and integrated with this pipeline to provide users with confidence that the reported results occur above chance over several randomized trials.

Like the context recognition suggestions, the accuracy of the T2F system can be improved by providing more training data to the learning model during the model fitting process. Since the current model was only trained on 928 text-function pairs, it is likely that expansion of the training set to a few thousand pairs would result in a significant

increase in the representation of potential responses. The accuracy and consistency of chatbots powered by T2F could provide more accurate responses to the more unstructured and unpredictable human text inputs that are likely to occur when users ask questions off the top of their head. The T2F system generalizes remarkably well to sentences with similar order, similar meaning, but synonymous wording. This is due to the discarding of word identity in favour of extracting generalized meaning that results from the use of a thesaurus. Further generalizability is conferred by the use of linear convolutional pre-processing on the flattened embeddings, which forces the network to identify a global polynomial pattern associated with the values of every dimension of each embedding.

To overcome the difficulty of manually curating thousands of text transcripts with a single person workforce, it may be possible to augment the data using previously reported NLP data augmentation methods (Bergmanis *et al.*, 2017; Jia and Liang, 2016; Kobayashi, 2018). Notably, data recombination postulated by Jia and Liang (2016) offers promise for better generalization to sentences of different order. The challenge with regards to proper data recombination in NLP is the creation of new sentences that are reordered versions of their originals, but still retain the same meaning. Unlike image data, where augmentation methods are abundant and in widespread use (Ding *et al.*, 2016; Han *et al.*, 2018; Zhong *et al.*, 2017), NLP data is much sparser and therefore the same data augmentation methods often discard too much data, or alter the original data to an extent that does not reflect the classification labels assigned to them. A simple option to enhance the generalizability of the model with respect to sentence order is the removal

of the "order" metric from the embedding dimensions, however this runs the risk of negatively affecting validation accuracy on sentences where order is semantically pertinent (such as "I did do this" vs. "Did I do this"). Lastly, it may be possible to apply generative adversarial networks (GAN) to simultaneously train the discriminative T2F system in addition to several text transcript generative models, which would ultimately result in a class-by-class augmentation of varying quality.

While the core demands of AiDA's T2F system would be for the carrying out of naturally-formatted user requests, there would be an expectation from many users that the chatbot have basic conversational capabilities. In order to expand AiDA's conversational capabilities, LSTM models may be implemented that are trained and validated on conversational response pairs such as those curated in the Cornell movie dialogs corpus (Danescu-Niculescu-Mizil and Lee, 2011) and the Searchqa dataset which includes question-answer pairs from encyclopedia-type sources such as Jeopardy! (Dunn et al., 2017). The core intention of the "non_functional" class of labels in the T2F dataset was to allow the bot to determine if the user's request does not require a functional response. Should this be the case, the T2F system will defer the judgement to a conversational LSTM to provide an organically-formulated conversational response.

4.4: Future directions for the AiDA platform

In the latest release of AiDA (version 0.1.4, http://www.aifive.tech/personal.php), the mindmap tool for idea visualization was shown to be capable for basic biological pathway analysis by connecting it internally to Pathway Commons data (Cerami *et al.*, 2010). Several features currently exist in the latest build that allow for simple pathway

analysis by asking AiDA to show genes on the mindmap. Every gene shown on the mindmap is automatically connected to other genes on the visualizer by performing a lookup to Pathway Commons data. Future work on this platform will involve expansions to the complexity of pathway analysis and visualization directed by the T2F system. Pathway analysis is currently a powerful method of identifying more nuanced determinants of disease that applies connectivity statistics to analyze networks of biological interactors. Visually, it is a supplement to the literature review process that helps researchers gain a better sense of how their research foci interact with other genes, chemicals, and biological processes.

If AiDA can be successfully completed as an end-to-end solution with the assistance of a development team in the future, there is reason to believe that the platform would have disruptive potential to the current research industry. It would reduce the time taken for even the most basic investigative tasks from several minutes or hours to a couple seconds. It would keep track of how, where, and when ideas are being searched, and attempt to assist the user in connecting those ideas together using several powerful AI-enabled dashboard applications. Should the accuracy of these deep learning models reach a human-like cognitive agility on specific tasks, we may learn to trust these automated tools as objective standards in the industry. The end result would be a raising of the general expectations of evidence in decision-making such that one or more extensive SLRs would be *required* to initiate a project due to the newfound simplicity of performing them. AiDA as a research tool will always remain freely accessible to its intended user base—the researcher—to assist in the generation of hypotheses of

increasingly profound impact. There are typically fewer personal advantages to offering complex technologies to the public *pro bono*, but the primary directive is not losing sight of what initially drove human innovation: the desire to understand the world around us, often for no personal benefit other than innate curiosity. As the AiDA chatbot solution continues to make algorithmic leaps and bounds using deep learning, over the last year to continue driving it forward as a powerful tool in evidence-based assisted research.

4.5: Translation of discoveries

The T2F system holds apparent translatability toward the development of effective, functional chatbots. This is because it simultaneously acts as an automated testing method to ensure that a large number of user requests can be predictably responded to, in addition to providing generalizability to unforeseen user requests. Often, modern chatbot applications fail in this regard, where user requests are funnelled through a subset of allowable responses in more primitive cases, and in more advanced cases fail to generalize to requests that require deep inference (for example, “I’m lost lol” should be responded to by asking the user if they would like a tutorial). The lack of generalizability in many cases is hinged on the enormous variability in word order, synonyms, spelling, and grammar that can be applied in conversation to indicate the same meaning. The T2F system drastically compresses this problem space by generalizing syntax through an expansive thesaurus with fast look-up, and an n-dimensional embedding system that can be standardized across training examples by linear convolution. While it still struggles to generalize for word order like other modern deep learning NLP methods, previous

postulations in *4.3*, if tested and implemented successfully, may assist in overcoming this obstacle.

The ranked list algorithm, which served to standardize feature extraction across datasets of diverse formatting and methodology, has immediate applicability to multi-dataset hypothesis generation. Currently, many NGS datasets are made publicly available at large cost to the donating institution(s). These datasets, while expensive and produced through extensive collaborative momentum, often do not comprise of enough samples to validate statistical hypotheses about their population characteristics. The ranked list algorithm provides a way to validate these hypotheses, made more generally about certain classes of samples, across many datasets. The justification for this method lies in the exponentially-decreasing likelihood that the same feature's prominence is due to chance as it repetitively appears in the extremes of many ranked lists. The idea that statistical stability can be conferred to repeatedly extreme ranked list entries is not new, having previously been discussed for the purposes of aggregating ranked lists of genes (Boulesteix and Slawski, 2009; Kolde *et al.*, 2012). The claim that this work makes, which is perhaps different from the canonical approach to gene list analysis, is the aggregation of many ranked lists analyzed in *different* ways for the purposes of fulfilling a hypothesis that is easy for a user to articulate verbally, but difficult to articulate mathematically.

Convolutional methods in NLP are presently poorly understood and have only recently been discussed in the literature (Britz, 2015; Kim, 2014; Zeng *et al.*, 2014). Technologies applied in this thesis, notably the collapsing of syntax and lexical information via a

thesaurus and the flattening of n-dimensional word embeddings and convolution of the resultant matrix, represent innovations in natural language processing that may be readily applied to enhance the performance of existing NLP software. This work corroborates the findings by Zeng *et al.* (2014) that linear convolution can be applied to word embeddings to improve model classification performance. Embedding generation through the use of semantic clustering tandem to extraction algorithms for innate characteristics of the text such as order index, plurality, and emphasis, removes the need for pre-trained word embedding vectors such as word2vec (Rong, 2014) and GloVe (Pennington *et al.*, 2014). The specific decision to apply linear convolution, as opposed to planar convolution, was due to the spatial irrelevance of the order of concatenation of embeddings in the 2D matrix. Linear embeddings reinforce learning of interdimensional characteristics within embeddings, as opposed to between embeddings, which enhances the generalizability of the model with regards to data sparsity and word representation.

4.6: Conclusion

Throughout the course of this thesis, a custom deep learning engine was implemented and validated for the purposes of carrying out user commands in intuitive ways and recognizing genetic and therapeutic contexts in the biomedical literature. A ranked list method for consolidating multi-dataset feature extraction was implemented to answer multifactorial user hypotheses, and expediate the process of data analysis and lead generation. Combined, the ranked list algorithms and deep learning engine were deployed in the form of the AiDA (Artificially-intelligent Desktop Assistant) platform,

which aims to transform the ways that research hypotheses are generated, investigated, and validated using collective evidence.

In Chapter 2, a variety of original algorithms, modifications to existing algorithms, and data handling methods were described, in addition to general dataset, hardware, and software characteristics. Much of the time invested in the completion of this thesis was placed in the development and automated testing of these algorithms. In Chapter 3, the deep learning engine was put to the test on multiple benchmark datasets commonly used by the data science community to validate machine learning models, including Digit-MNIST, Fashion-MNIST, and the CIFAR-10 set. While these datasets comprised of images, the transferability of the network's successes in these tasks became apparent when similar convolutional networks were applied in a new way to natural language processing. These networks performed similarly well on NLP examples, generalizing particularly well in the case of genetic and therapeutic context recognition due to the training set sizes. Ranked list algorithms were applied to generate leads for drug discovery in AML by searching for genes that were highly-expressed in patients with poor prognosis and in LSC+ samples, while simultaneously being low in normal hematopoietic tissue and primitive cell fractions. The characteristics of these leads were investigated, revealing that AML mutational hotspots correlated in incidence with expression of high-ranking leads. Furthermore, GSEA was performed to reveal that AML and LSC-related gene signatures were enriched at both extremes of the consolidated ranked list. The genetic and therapeutic context models were then applied

to mine the literature for potential therapeutics targeting the leads generated by the ranked list method.

Overall, the evidence presented was sufficient to support the claim that the research investigation process can be automated end-to-end by computational methods. While the accuracy of the methods demonstrated will require improvement through continual tuning and training data curation, the generalizability and relatively high accuracy of NLP models show that it is in fact possible to automate lead generation to some extent.

References

Abdullah, Wasi. "Making a Stand Alone Executable from a Python Script Using PyInstaller." *Medium*, 5 Oct. 2017, https://medium.com/dreamcatcher-its-blog/making-an-stand-alone-executable-from-a-python-script-using-pyinstaller-d1df9170e263.

Ågerstam, Helena, et al. "Antibodies Targeting Human IL1RAP (IL1R3) Show Therapeutic Effects in Xenograft Models of Acute Myeloid Leukemia." *Proceedings of the National Academy of Sciences*, vol. 112, no. 34, 2015, pp. 10786–91.

Alter, Blanche P. "Fanconi Anemia and the Development of Leukemia." *Best Practice & Research Clinical Haematology*, vol. 27, no. 3–4, 2014, pp. 214–21.

Amtmann, E. "The Antiviral, Antitumoural Xanthate D609 Is a Competitive Inhibitor of Phosphatidylcholine-Specific Phospholipase C." *Drugs under Experimental and Clinical Research*, vol. 22, no. 6, 1996, pp. 287–94.

Araújo, Teresa, et al. "Classification of Breast Cancer Histology Images Using Convolutional Neural Networks." *PloS One*, vol. 12, no. 6, 2017, p. e0177544.

Arber, Daniel A., et al. "The 2016 Revision to the World Health Organization Classification of Myeloid Neoplasms and Acute Leukemia." *Blood*, vol. 127, no. 20, 2016, pp. 2391–405.

Archbold, Julia K., et al. "Structural Insights into RAMP Modification of Secretin Family G Protein-Coupled Receptors: Implications for Drug Development." *Trends in Pharmacological Sciences*, vol. 32, no. 10, 2011, pp. 591–600.

Ashburner, Michael, et al. "Gene Ontology: Tool for the Unification of Biology." *Nature Genetics*, vol. 25, no. 1, 2000, p. 25.

Askmyr, Maria, et al. "Selective Killing of Candidate AML Stem Cells by Antibody Targeting of IL1RAP." *Blood*, vol. 121, no. 18, 2013, pp. 3709–13.

Auerbach, Arleen D., and RG Allen. "Leukemia and Preleukemia in Fanconi Anemia Patients: A Review of the Literature and Report of the International Fanconi Anemia Registry." *Cancer Genetics and Cytogenetics*, vol. 51, no. 1, 1991, pp. 1–12.

Bagger, Frederik Otzen, et al. "BloodSpot: A Database of Gene Expression Profiles and Transcriptional Programs for Healthy and Malignant Haematopoiesis." *Nucleic Acids Research*, vol. 44, no. D1, 2015, pp. D917–24.

Bateman, Alain R., et al. "Importance of Collection in Gene Set Enrichment Analysis of Drug Response in Cancer Cell Lines." *Scientific Reports*, vol. 4, 2014, p. 4092.

Batsukh, Tserendulam, et al. "Identification and Characterization of FAM124B as a Novel Component of a CHD7 and CHD8 Containing Complex." *PLoS ONE*, edited

by Christoph Englert, vol. 7, no. 12, Dec. 2012, p. e52640. *Crossref*, doi:10.1371/journal.pone.0052640.

Bello, Irwan, et al. *Neural Optimizer Search with Reinforcement Learning*. JMLR. org, 2017, pp. 459–68.

Benjamini, Yoav, et al. "Controlling the False Discovery Rate in Behavior Genetics Research." *Behavioural Brain Research*, vol. 125, no. 1–2, 2001, pp. 279–84.

Bergmanis, Toms, et al. *Training Data Augmentation for Low-Resource Morphological Inflection*. 2017, pp. 31–39.

Bernt, Kathrin M., et al. "MLL-Rearranged Leukemia Is Dependent on Aberrant H3K79 Methylation by DOT1L." *Cancer Cell*, vol. 20, no. 1, 2011, pp. 66–78.

Bezrodnik, L., et al. "Follicular Bronchiolitis as Phenotype Associated with CD25 Deficiency: CD25 Deficiency." *Clinical & Experimental Immunology*, vol. 175, no. 2, Feb. 2014, pp. 227–34. *Crossref*, doi:10.1111/cei.12214.

Bielinski, Suzette J., et al. *Preemptive Genotyping for Personalized Medicine: Design of the Right Drug, Right Dose, Right Time—Using Genomic Data to Individualize Treatment Protocol*. Vol. 89, Elsevier, 2014, pp. 25–33.

Boeuf, Stephane, et al. "Enhanced ITM2A Expression Inhibits Chondrogenic Differentiation of Mesenchymal Stem Cells." *Differentiation*, vol. 78, no. 2–3, 2009, pp. 108–15.

Bonnet, Dominique, and John E. Dick. "Human Acute Myeloid Leukemia Is Organized as a Hierarchy That Originates from a Primitive Hematopoietic Cell." *Nature Medicine*, vol. 3, no. 7, 1997, p. 730.

Boulesteix, Anne-Laure, and Martin Slawski. "Stability and Aggregation of Ranked Gene Lists." *Briefings in Bioinformatics*, vol. 10, no. 5, 2009, pp. 556–68.

Brandtzaeg, Petter Bae, and Asbjørn Følstad. "Why People Use Chatbots." *Internet Science*, edited by Ioannis Kompatsiaris et al., vol. 10673, Springer International Publishing, 2017, pp. 377–92. *Crossref*, doi:10.1007/978-3-319-70284-1_30.

Britz, Denny. "Understanding Convolutional Neural Networks for NLP." *Http://Www.Wildml.Com/2015/11/Understanding-Convolutional-Neuralnetworks-for-Nlp/*, 2015.

Brogger, Jan. "Reporting of Systematic Reviews: Better Software Required." *PLoS Medicine*, vol. 4, no. 6, June 2007, p. e225. *Crossref*, doi:10.1371/journal.pmed.0040225.

Bullers, Krystal, et al. "It Takes Longer than You Think: Librarian Time Spent on Systematic Review Tasks." *Journal of the Medical Library Association*, vol. 106, no. 2, Apr. 2018. *Crossref*, doi:10.5195/JMLA.2018.323.

Büschges, Rainer, et al. "Amplification and Expression of Cyclin D Genes (CCND1 CCND2 and CCND3) in Human Malignant Gliomas." *Brain Pathology*, vol. 9, no. 3, 1999, pp. 435–42.

Cancer Genome Atlas Research Network. "Genomic and Epigenomic Landscapes of Adult de Novo Acute Myeloid Leukemia." *New England Journal of Medicine*, vol. 368, no. 22, 2013, pp. 2059–74.

Cerami, Ethan G., et al. "Pathway Commons, a Web Resource for Biological Pathway Data." *Nucleic Acids Research*, vol. 39, no. suppl_1, 2010, pp. D685–90.

Chapman, Andrea L., et al. "Semi-Automating the Manual Literature Search for Systematic Reviews Increases Efficiency." *Health Information & Libraries Journal*, vol. 27, no. 1, Mar. 2010, pp. 22–27. *Crossref*, doi:10.1111/j.1471-1842.2009.00865.x.

Chen, Dong-Hui, et al. "A Novel Mutation in FHL1 in a Family with X-Linked Scapuloperoneal Myopathy: Phenotypic Spectrum and Structural Study of FHL1 Mutations." *Journal of the Neurological Sciences*, vol. 296, no. 1–2, 2010, pp. 22–29.

Chen, Sheng, et al. "Non-Linear System Identification Using Neural Networks." *International Journal of Control*, vol. 51, no. 6, 1990, pp. 1191–214.

Chen, Yen-Chen, et al. "Cancer Adjuvant Chemotherapy Strategic Classification by Artificial Neural Network with Gene Expression Data: An Example for Non-Small Cell Lung Cancer." *Journal of Biomedical Informatics*, vol. 56, 2015, pp. 1–7.

Cheng, Donavan T., et al. "Memorial Sloan Kettering-Integrated Mutation Profiling of Actionable Cancer Targets (MSK-IMPACT): A Hybridization Capture-Based next-Generation Sequencing Clinical Assay for Solid Tumor Molecular Oncology." *The Journal of Molecular Diagnostics*, vol. 17, no. 3, 2015, pp. 251–64.

CompareCamp. "IBM Watson Review: Pricing, Pros, Cons & Features | CompareCamp.Com." *CompareCamp*, 10 Jan. 2019, http://comparecamp.com/ibm-watson-review-pricing-pros-cons-features/.

Corces, M. Ryan, et al. "Lineage-Specific and Single-Cell Chromatin Accessibility Charts Human Hematopoiesis and Leukemia Evolution." *Nature Genetics*, vol. 48, no. 10, 2016, p. 1193.

Cortes, Jorge E., et al. *Crenolanib Besylate, a Type I Pan-FLT3 Inhibitor, to Demonstrate Clinical Activity in Multiply Relapsed FLT3-ITD and D835 AML*. 2016.

Cottrell, Catherine E., et al. "Validation of a Next-Generation Sequencing Assay for Clinical Molecular Oncology." *The Journal of Molecular Diagnostics*, vol. 16, no. 1, 2014, pp. 89–105.

Cybenko, George. “Approximation by Superpositions of a Sigmoidal Function.” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, 1989, pp. 303–14.

Dackor, Ryan, et al. “Receptor Activity-Modifying Proteins 2 and 3 Have Distinct Physiological Functions from Embryogenesis to Old Age.” *Journal of Biological Chemistry*, vol. 282, no. 25, 2007, pp. 18094–99.

Dahl, George E., et al. *Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout*. IEEE, 2013, pp. 8609–13.

Dalby, Kevin N., et al. “Identification of Regulatory Phosphorylation Sites in Mitogen-Activated Protein Kinase (MAPK)-Activated Protein Kinase-1a/P90 Rsk That Are Inducible by MAPK.” *Journal of Biological Chemistry*, vol. 273, no. 3, 1998, pp. 1496–505.

Danescu-Niculescu-Mizil, Cristian, and Lillian Lee. *Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs*. Association for Computational Linguistics, 2011, pp. 76–87.

Darken, Christian, and John E. Moody. *Note on Learning Rate Schedules for Stochastic Optimization*. 1991, pp. 832–38.

De Magalhães, João Pedro, et al. “Next-Generation Sequencing in Aging Research: Emerging Applications, Problems, Pitfalls and Possible Solutions.” *Ageing Research Reviews*, vol. 9, no. 3, 2010, pp. 315–23.

Delen, Dursun, and Martin D. Crossland. “Seeding the Survey and Analysis of Research Literature with Text Mining.” *Expert Systems with Applications*, vol. 34, no. 3, 2008, pp. 1707–20.

Deng, Li. “The MNIST Database of Handwritten Digit Images for Machine Learning Research.” *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012, pp. 141–42.

Ding, Jun, et al. “Convolutional Neural Network with Data Augmentation for SAR Target Recognition.” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 3, 2016, pp. 364–68.

Dinu, Irina, et al. “Improving Gene Set Analysis of Microarray Data by SAM-GS.” *BMC Bioinformatics*, vol. 8, no. 1, 2007, p. 242.

Döhner, Hartmut, et al. “Acute Myeloid Leukemia.” *New England Journal of Medicine*, vol. 373, no. 12, 2015, pp. 1136–52.

Domhan, Tobias, et al. *Speeding up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves*. 2015.

Dove, Graham, et al. *Ux Design Innovation: Challenges for Working with Machine Learning as a Design Material*. ACM, 2017, pp. 278–88.

Dozat, Timothy. *Incorporating Nesterov Momentum into Adam*. Vol. 107, 2016, pp. 1–3.

Du, Wei, et al. "The Fanconi Anemia Pathway Controls Oncogenic Response in Hematopoietic Stem and Progenitor Cells by Regulating PRMT5-Mediated P53 Arginine Methylation." *Oncotarget*, vol. 7, no. 37, 2016, p. 60005.

Du, Wen, et al. "High IL2RA MRNA Expression Is an Independent Adverse Prognostic Biomarker in Core Binding Factor and Intermediate-Risk Acute Myeloid Leukemia." *Journal of Translational Medicine*, vol. 17, no. 1, Dec. 2019. *Crossref*, doi:10.1186/s12967-019-1926-z.

Dunn, Matthew, et al. "Searchqa: A New Q&a Dataset Augmented with Context from a Search Engine." *ArXiv Preprint ArXiv:1704.05179*, 2017.

Dunne, Rob A., and Norm A. Campbell. *On the Pairing of the Softmax Activation and Cross-Entropy Penalty Functions and the Derivation of the Softmax Activation Function*. Vol. 181, Citeseer, 1997, p. 185.

Dyba, Tore, et al. "Evidence-Based Software Engineering for Practitioners." *IEEE Software*, vol. 22, no. 1, 2005, pp. 58–65.

Eppert, Kolja, et al. "Stem Cell Gene Expression Programs Influence Clinical Outcome in Human Leukemia." *Nature Medicine*, vol. 17, no. 9, 2011, p. 1086.

Esteva, Andre, et al. "Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks." *Nature*, vol. 542, no. 7639, 2017, p. 115.

Evidence Partners. *DistillerSR Systematic Review Software*. Evidence Partners, 2011, https://www.evidencepartners.com/products/distillersr-systematic-review-software/.

Exton, JH. "Signaling through Phosphatidylcholine Breakdown." *Journal of Biological Chemistry*, vol. 265, no. 1, 1990, pp. 1–4.

Fang, Che, et al. "Effect of Ribavirin, Levovirin and Viramidine on Liver Toxicological Gene Expression in Rats." *Journal of Applied Toxicology*, vol. 23, no. 6, Nov. 2003, pp. 453–59. *Crossref*, doi:10.1002/jat.938.

Farley, B., and W. Clark. "Simulation of Self-Organizing Systems by Digital Computer." *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, Sept. 1954, pp. 76–84. *Crossref*, doi:10.1109/TIT.1954.1057468.

Fattahi, Rana, et al. "Pancreatic Diffusion-weighted Imaging (DWI): Comparison between Mass-forming Focal Pancreatitis (FP), Pancreatic Cancer (PC), and Normal Pancreas." *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 29, no. 2, 2009, pp. 350–56.

Fernández-Sáez, Ana M., et al. *SLR-Tool: A Tool for Performing Systematic Literature Reviews*. 2010, pp. 157–66.

Foon, Kenneth A. “Immunologic Classification of Leukemia and Lymphoma.” *Blood*, vol. 68, no. 1, 1986, pp. 1–31.

Franceschini, Andrea, et al. “STRING v9. 1: Protein-Protein Interaction Networks, with Increased Coverage and Integration.” *Nucleic Acids Research*, vol. 41, no. D1, 2012, pp. D808–15.

Gao, Jianjiong, et al. “Integrative Analysis of Complex Cancer Genomics and Clinical Profiles Using the CBioPortal.” *Sci. Signal.*, vol. 6, no. 269, 2013, pp. pl1–pl1.

Gentles, Andrew J. “Association of a Leukemic Stem Cell Gene Expression Signature With Clinical Outcomes in Acute Myeloid Leukemia.” *JAMA*, vol. 304, no. 24, Dec. 2010, p. 2706. *Crossref*, doi:10.1001/jama.2010.1862.

Giustacchini, Alice, et al. “Single-Cell Transcriptomics Uncovers Distinct Molecular Signatures of Stem Cells in Chronic Myeloid Leukemia.” *Nature Medicine*, vol. 23, no. 6, 2017, p. 692.

Glorot, Xavier, et al. *Deep Sparse Rectifier Neural Networks*. 2011, pp. 315–23.

Goble, Carole, and Robert Stevens. “State of the Nation in Data Integration for Bioinformatics.” *Journal of Biomedical Informatics*, vol. 41, no. 5, Oct. 2008, pp. 687–93. *CrossRef*, doi:10.1016/j.jbi.2008.01.008.

Gomaa, Wael H., and Aly A. Fahmy. “Short Answer Grading Using String Similarity and Corpus-Based Similarity.” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 3, no. 11, 2012.

Goudy, Kevin, et al. “Human IL2RA Null Mutation Mediates Immunodeficiency with Lymphoproliferation and Autoimmunity.” *Clinical Immunology*, vol. 146, no. 3, Mar. 2013, pp. 248–61. *Crossref*, doi:10.1016/j.clim.2013.01.004.

Gray, Colin M., et al. *Flow of Competence in UX Design Practice*. ACM, 2015, pp. 3285–94.

Green, Timothy. *IBM’s Cognitive Solutions Sales Slumped. What Happened?* 17 Oct. 2018, https://www.fool.com/investing/2018/10/17/ibms-cognitive-solutions-sales-slumped-what-happen.aspx.

Guan, Yan-Fang, et al. “Application of Next-Generation Sequencing in Clinical Oncology to Advance Personalized Treatment of Cancer.” *Chinese Journal of Cancer*, vol. 31, no. 10, 2012, p. 463.

Han, Dongmei, et al. “A New Image Classification Method Using CNN Transfer Learning and Web Data Augmentation.” *Expert Systems with Applications*, vol. 95, 2018, pp. 43–56.

Hanash, Samir M., et al. “Mining the Plasma Proteome for Cancer Biomarkers.” *Nature*, vol. 452, no. 7187, 2008, p. 571.

Hardy, John, and Bart De Strooper. "Alzheimer's Disease: Where next for Anti-Amyloid Therapies?" *Brain*, vol. 140, no. 4, Mar. 2017, pp. 853–55, doi:10.1093/brain/awx059.

Hautamäki, Ville, et al. *Automatic Regularization of Cross-Entropy Cost for Speaker Recognition Fusion*. 2013.

Hebb, Donald Olding. *The Organization of Behavior: A Neuropsychological Theory*. Science Editions, 1962.

Hecht-Nielsen, Robert. "Theory of the Backpropagation Neural Network**Based on 'Nonindent' by Robert Hecht-Nielsen, Which Appeared in Proceedings of the International Joint Conference on Neural Networks 1, 593–611, June 1989. © 1989 IEEE." *Neural Networks for Perception*, Elsevier, 1992, pp. 65–93. *Crossref*, doi:10.1016/B978-0-12-741252-8.50010-8.

Hehlmann, Rüdiger, et al. "Chronic Myeloid Leukaemia." *The Lancet*, vol. 370, no. 9584, 2007, pp. 342–50.

Hennessy, Bryan T., et al. "Exploiting the PI3K/AKT Pathway for Cancer Drug Discovery." *Nature Reviews Drug Discovery*, vol. 4, no. 12, 2005, p. 988.

Higgins, Julian PT, and Sally Green. *Cochrane Handbook for Systematic Reviews of Interventions*. 2008.

Ho, Tzu-Chieh, et al. "Evolution of Acute Myelogenous Leukemia Stem Cell Properties after Treatment and Progression." *Blood*, vol. 128, no. 13, 2016, pp. 1671–78.

Hochreiter, Sepp. "The Vanishing Gradient Problem during Learning Recurrent Neural Nets and Problem Solutions." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, 1998, pp. 107–16.

Holi, Ganga, and Divya K. Jain. "Convolutional Neural Network Approach for Extraction and Recognition of Digits from Bank Cheque Images." *Emerging Research in Electronics, Computer Science and Technology*, Springer, 2019, pp. 331–41.

Hope, Kristin J., et al. "Acute Myeloid Leukemia Originates from a Hierarchy of Leukemic Stem Cell Classes That Differ in Self-Renewal Capacity." *Nature Immunology*, vol. 5, no. 7, 2004, p. 738.

Horiike, S., et al. "Tandem Duplications of the FLT3 Receptor Gene Are Associated with Leukemic Transformation of Myelodysplasia." *Leukemia*, vol. 11, no. 9, 1997, p. 1442.

Horvath, Lisa G., et al. "Frequent Loss of Estrogen Receptor-β Expression in Prostate Cancer." *Cancer Research*, vol. 61, no. 14, 2001, pp. 5331–35.

Huang, Sui. "Gene Expression Profiling, Genetic Networks, and Cellular States: An Integrating Concept for Tumorigenesis and Drug Discovery." *Journal of Molecular Medicine*, vol. 77, no. 6, 1999, pp. 469–80.

Huang, Wenyi, and Jack W. Stokes. *MtNet: A Multi-Task Neural Network for Dynamic Malware Classification*. Springer, 2016, pp. 399–418.

Ihaka, Ross, and Robert Gentleman. "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, vol. 5, no. 3, 1996, pp. 299–314.

Ivakhnenko, Alekseĭ Grigor′evich, and Valentin Grigor′evich Lapa. *Cybernetics and Forecasting Techniques*. 1967.

Ivanisenko, Vladimir A., et al. "ANDSystem: An Associative Network Discovery System for Automated Literature Mining in the Field of Biology." *BMC Systems Biology*, vol. 9, no. 2, 2015, p. S2.

Jarvis, Kevin. *IBM Watson Commerce Review: Pros, Cons and Pricing Comparison 2019*. Jan. 2019, https://www.jarviscole.com/blog/2018/12/ibm-watson-commerce-review.

Jia, Robin, and Percy Liang. "Data Recombination for Neural Semantic Parsing." *ArXiv Preprint ArXiv:1606.03622*, 2016.

Joenje, Hans, and Ketan J. Patel. "The Emerging Genetic and Molecular Basis of Fanconi Anaemia." *Nature Reviews Genetics*, vol. 2, no. 6, 2001, p. 446.

Jonnalagadda, Siddhartha R., et al. "Automating Data Extraction in Systematic Reviews: A Systematic Review." *Systematic Reviews*, vol. 4, no. 1, Dec. 2015. *Crossref*, doi:10.1186/s13643-015-0066-7.

Jordan, Nicole Vincent, et al. "HER2 Expression Identifies Dynamic Functional States within Circulating Breast Cancer Cells." *Nature*, vol. 537, no. 7618, 2016, p. 102.

Jung, Heechul, et al. *Joint Fine-Tuning in Deep Neural Networks for Facial Expression Recognition*. 2015, pp. 2983–91.

Jung, Jin-Gyoung, et al. "Ovarian Cancer Chemoresistance Relies on the Stem Cell Reprogramming Factor PBX1." *Cancer Research*, vol. 76, no. 21, 2016, pp. 6351–61.

Kametani, Fuyuki, and Masato Hasegawa. "Reconsideration of Amyloid Hypothesis and Tau Hypothesis in Alzheimer's Disease." *Frontiers in Neuroscience*, vol. 12, 2018, p. 25, doi:10.3389/fnins.2018.00025.

Kamisawa, Terumi, et al. "Strategy for Differentiating Autoimmune Pancreatitis from Pancreatic Cancer." *Gastrointestinal Endoscopy*, vol. 67, no. 5, 2008, p. AB232.

Keskar, Nitish Shirish, and Richard Socher. "Improving Generalization Performance by Switching from Adam to Sgd." *ArXiv Preprint ArXiv:1712.07628*, 2017.

Kikushige, Yoshikane, et al. "Self-Renewing Hematopoietic Stem Cell Is the Primary Target in Pathogenesis of Human Chronic Lymphocytic Leukemia." *Cancer Cell*, vol. 20, no. 2, 2011, pp. 246–59.

Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." *ArXiv Preprint ArXiv:1408.5882*, 2014.

Kindler, Thomas, et al. "FLT3 as a Therapeutic Target in AML: Still Challenging after All These Years." *Blood*, vol. 116, no. 24, 2010, pp. 5089–102.

Kingma, Diederik P., and Jimmy Ba. "Adam: A Method for Stochastic Optimization." *ArXiv Preprint ArXiv:1412.6980*, 2014.

Kline, Douglas M., and Victor L. Berardi. "Revisiting Squared-Error and Cross-Entropy Functions for Training Neural Network Classifiers." *Neural Computing & Applications*, vol. 14, no. 4, 2005, pp. 310–18.

Kobayashi, Sosuke. "Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations." *ArXiv Preprint ArXiv:1805.06201*, 2018.

Kolde, Raivo, et al. "Robust Rank Aggregation for Gene List Integration and Meta-Analysis." *Bioinformatics*, vol. 28, no. 4, 2012, pp. 573–80.

Kotini, Andriana G., et al. "Stage-Specific Human Induced Pluripotent Stem Cells Map the Progression of Myeloid Transformation to Transplantable Leukemia." *Cell Stem Cell*, vol. 20, no. 3, 2017, pp. 315–28.

Kottaridis, Panagiotis D., et al. "The Presence of a FLT3 Internal Tandem Duplication in Patients with Acute Myeloid Leukemia (AML) Adds Important Prognostic Information to Cytogenetic Risk Group and Response to the First Cycle of Chemotherapy: Analysis of 854 Patients from the United Kingdom Medical Research Council AML 10 and 12 Trials." *Blood*, vol. 98, no. 6, 2001, pp. 1752–59.

Krähenbühl, Philipp, et al. "Data-Dependent Initializations of Convolutional Neural Networks." *ArXiv Preprint ArXiv:1511.06856*, 2015.

Kreitman, Robert J., et al. "Phase I Trial of Recombinant Immunotoxin Anti-Tac (Fv)-PE38 (LMB-2) in Patients with Hematologic Malignancies." *Journal of Clinical Oncology*, vol. 18, no. 8, 2000, pp. 1622–36.

---. "Recombinant Immunotoxins Containing Truncated Bacterial Toxins for the Treatment of Hematologic Malignancies." *BioDrugs*, vol. 23, no. 1, 2009, pp. 1–13.

Kreitman, Robert J., and Ira Pastan. "TargetingPseudomonasexotoxin to Hematologic Malignancies." *Seminars in Cancer Biology*, vol. 6, no. 5, Oct. 1995, pp. 297–306. *Crossref*, doi:10.1006/scbi.1995.0038.

Kreso, Antonija, and John E. Dick. "Evolution of the Cancer Stem Cell Model." *Cell Stem Cell*, vol. 14, no. 3, 2014, pp. 275–91.

Krizhevsky, Alex, and Geoffrey Hinton. *Learning Multiple Layers of Features from Tiny Images*. Citeseer, 2009.

Kumar, Anurag, and Dinei Florencio. "Speech Enhancement in Multiple-Noise Conditions Using Deep Neural Networks." *ArXiv Preprint ArXiv:1605.02427*, 2016.

Labaer, Joshua. "Mining the Literature and Large Datasets." *Nature Biotechnology*, vol. 21, no. 9, 2003, p. 976.

Lawrence, Steve, et al. "Face Recognition: A Convolutional Neural-Network Approach." *IEEE Transactions on Neural Networks*, vol. 8, no. 1, 1997, pp. 98–113.

Leal, Letícia F., et al. "Reproducibility of the NanoString 22-gene Molecular Subgroup Assay for Improved Prognostic Prediction of Medulloblastoma." *Neuropathology*, vol. 38, no. 5, 2018, pp. 475–83.

Lebofsky, Ronald, et al. "Circulating Tumor DNA as a Non-invasive Substitute to Metastasis Biopsy for Tumor Genotyping and Personalized Medicine in a Prospective Trial across All Tumor Types." *Molecular Oncology*, vol. 9, no. 4, 2015, pp. 783–90.

LeCun, Yann, et al. "Deep Learning." *Nature*, vol. 521, no. 7553, May 2015, pp. 436–44. *CrossRef*, doi:10.1038/nature14539.

Leshno, Moshe, et al. "Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate Any Function." *Neural Networks*, vol. 6, no. 6, Jan. 1993, pp. 861–67. *Crossref*, doi:10.1016/S0893-6080(05)80131-5.

Li, Chenwei, David G. Heidt, Piero Dalerba, et al. "Identification of Pancreatic Cancer Stem Cells." *Cancer Research*, vol. 67, no. 3, 2007, pp. 1030–37.

Li, Rita Yi Man, Simon Fong, and Kyle Weng Sang Chong. "Forecasting the REITs and Stock Indices: Group Method of Data Handling Neural Network Approach." *Pacific Rim Property Research Journal*, vol. 23, no. 2, 2017, pp. 123–60.

Li, Sheng, Francine E. Garrett-Bakelman, et al. "Distinct Evolution and Dynamics of Epigenetic and Genetic Heterogeneity in Acute Myeloid Leukemia." *Nature Medicine*, vol. 22, no. 7, 2016, p. 792.

Li, Sheng, Christopher E. Mason, et al. "Genetic and Epigenetic Heterogeneity in Acute Myeloid Leukemia." *Current Opinion in Genetics & Development*, vol. 36, 2016, pp. 100–06.

Libbrecht, Maxwell W., and William Stafford Noble. "Machine Learning Applications in Genetics and Genomics." *Nature Reviews Genetics*, vol. 16, no. 6, 2015, p. 321.

Liberzon, Arthur, et al. "Molecular Signatures Database (MSigDB) 3.0." *Bioinformatics*, vol. 27, no. 12, 2011, pp. 1739–40.

Ligowski, Lukasz, and Witold Rudnicki. *An Efficient Implementation of Smith Waterman Algorithm on GPU Using CUDA, for Massively Parallel Scanning of Sequence Databases*. IEEE, 2009, pp. 1–8.

Lilljebjörn, Henrik, et al. "RNA-Seq Identifies Clinically Relevant Fusion Genes in Leukemia Including a Novel MEF2D/CSF1R Fusion Responsive to Imatinib." *Leukemia*, vol. 28, no. 4, 2014, p. 977.

Lin, Ling, et al. "The Ser/Thr Kinase P90RSK Promotes Kidney Fibrosis by Modulating Fibroblast–Epithelial Crosstalk." *Journal of Biological Chemistry*, vol. 294, no. 25, June 2019, pp. 9901–10. *Crossref*, doi:10.1074/jbc.RA119.007904.

Lin, W. "Expression and Function of the HSD-3.8 Gene Encoding a Testis-Specific Protein." *Molecular Human Reproduction*, vol. 7, no. 9, Sept. 2001, pp. 811–18. *Crossref*, doi:10.1093/molehr/7.9.811.

Lingel, Andreas, et al. "Structure of IL-33 and Its Interaction with the ST2 and IL-1RAcP Receptors—Insight into Heterotrimeric IL-1 Signaling Complexes." *Structure*, vol. 17, no. 10, Oct. 2009, pp. 1398–410. *Crossref*, doi:10.1016/j.str.2009.08.009.

Liu, B. H., et al. "Identification of Unique and Common Low Abundance Tumour-Specific Transcripts by Suppression Subtractive Hybridization and Oligonucleotide Probe Array Analysis." *Oncogene*, vol. 27, no. 29, July 2008, pp. 4128–36. *Crossref*, doi:10.1038/onc.2008.50.

Liu, Jane Jijun, et al. "Multiclass Cancer Classification and Biomarker Discovery Using GA-Based Algorithms." *Bioinformatics*, vol. 21, no. 11, 2005, pp. 2691–97.

Liu, Zhehui, et al. "MiR-592 Inhibited Cell Proliferation of Human Colorectal Cancer Cells by Suppressing of CCND3 Expression." *International Journal of Clinical and Experimental Medicine*, vol. 8, no. 3, 2015, p. 3490.

Loshchilov, Ilya, and Frank Hutter. "CMA-ES for Hyperparameter Optimization of Deep Neural Networks." *ArXiv Preprint ArXiv:1604.07269*, 2016.

Maas, Andrew L., et al. *Rectifier Nonlinearities Improve Neural Network Acoustic Models*. Vol. 30, 2013, p. 3.

MacKay, David JC. "Comparison of Approximate Methods for Handling Hyperparameters." *Neural Computation*, vol. 11, no. 5, 1999, pp. 1035–68.

---. "Hyperparameters: Optimize, or Integrate Out?" *Maximum Entropy and Bayesian Methods*, Springer, 1996, pp. 43–59.

Makin, Simon. "The Amyloid Hypothesis on Trial." *Nature*, vol. 559, no. 7715, July 2018, pp. S4–7. *Crossref*, doi:10.1038/d41586-018-05719-4.

McCulloch, Warren S., and Walter Pitts. “A Logical Calculus of the Ideas Immanent in Nervous Activity.” *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, 1943, pp. 115–33.

Merelli, Ivan, et al. “Managing, Analysing, and Integrating Big Data in Medical Bioinformatics: Open Problems and Future Perspectives.” *BioMed Research International*, vol. 2014, 2014.

Meshinchi, Soheil, et al. “Clinical Implications of FLT3 Mutations in Pediatric AML.” *Blood*, vol. 108, no. 12, 2006, pp. 3654–61.

Mitchell, Kelly, et al. “IL1RAP Potentiates Multiple Oncogenic Signaling Pathways in AML.” *Journal of Experimental Medicine*, vol. 215, no. 6, 2018, pp. 1709–27.

Morgan, MJ, and AJA Madgwick. “The LIM Proteins FHL1 and FHL3 Are Expressed Differently in Skeletal Muscle.” *Biochemical and Biophysical Research Communications*, vol. 255, no. 2, 1999, pp. 245–50.

Muir, Paul, et al. “The Real Cost of Sequencing: Scaling Computation to Keep Pace with Data Generation.” *Genome Biology*, vol. 17, no. 1, 2016, p. 53.

Murohashi, M., et al. “Gene Set Enrichment Analysis Provides Insight into Novel Signalling Pathways in Breast Cancer Stem Cells.” *British Journal of Cancer*, vol. 102, no. 1, 2010, p. 206.

Musso, G., et al. “Generating and Evaluating a Ranked Candidate Gene List for Potential Vertebrate Heart Field Regulators.” *Genomics Data*, vol. 6, 2015, pp. 199–201.

Naderi, A., et al. “A Gene-Expression Signature to Predict Survival in Breast Cancer across Independent Data Sets.” *Oncogene*, vol. 26, no. 10, Mar. 2007, pp. 1507–16. *Crossref*, doi:10.1038/sj.onc.1209920.

Nair, Vinod, and Geoffrey E. Hinton. *Rectified Linear Units Improve Restricted Boltzmann Machines*. 2010, pp. 807–14.

Nalçakan, Yağız, and Tolga Ensari. *Decision of Neural Networks Hyperparameters with a Population-Based Algorithm*. Springer, 2018, pp. 276–81.

Ng, Stanley W. K., et al. “A 17-Gene Stemness Score for Rapid Determination of Risk in Acute Leukaemia.” *Nature*, vol. 540, no. 7633, 15 2016, pp. 433–37. *PubMed*, doi:10.1038/nature20598.

Onda, Masanori, Mark Willingham, et al. “Inhibition of TNF-α Produced by Kupffer Cells Protects against the Nonspecific Liver Toxicity of Immunotoxin Anti-Tac (Fv)-PE38, LMB-2.” *The Journal of Immunology*, vol. 165, no. 12, 2000, pp. 7150–56.

Onda, Masanori, Robert J. Kreitman, et al. “Reduction of the Nonspecific Animal Toxicity of Anti-Tac (Fv)-PE38 by Mutations in the Framework Regions of the Fv

Which Lower the Isoelectric Point." *The Journal of Immunology*, vol. 163, no. 11, 1999, pp. 6072–77.

Ooi, CH, and Patrick Tan. "Genetic Algorithms Applied to Multi-Class Prediction for the Analysis of Gene Expression Data." *Bioinformatics*, vol. 19, no. 1, 2003, pp. 37–44.

Øvad, Tina, and Lars Bo Larsen. *The Prevalence of UX Design in Agile Development Processes in Industry*. IEEE, 2015, pp. 40–49.

Pantel, Austin, et al. "Direct Type I IFN but Not MDA5/TLR3 Activation of Dendritic Cells Is Required for Maturation and Metabolic Shift to Glycolysis after Poly IC Stimulation." *PLoS Biology*, vol. 12, no. 1, 2014, p. e1001759.

Park, Yeon Hee, et al. "A Seven-gene Signature Can Predict Distant Recurrence in Patients with Triple-negative Breast Cancers Who Receive Adjuvant Chemotherapy Following Surgery." *International Journal of Cancer*, vol. 136, no. 8, 2015, pp. 1976–84.

Patel, Nishma, et al. "Cost Analysis of Standard Sanger Sequencing versus next Generation Sequencing in the ICONIC Study." *The Lancet*, vol. 388, 2016, p. S86.

Payan, Adrien, and Giovanni Montana. "Predicting Alzheimer's Disease: A Neuroimaging Study with 3D Convolutional Neural Networks." *ArXiv Preprint ArXiv:1502.02506*, 2015.

Pearson, William R. "Searching Protein Sequence Libraries: Comparison of the Sensitivity and Selectivity of the Smith-Waterman and FASTA Algorithms." *Genomics*, vol. 11, no. 3, 1991, pp. 635–50.

Pelish, Henry E., et al. "Mediator Kinase Inhibition Further Activates Super-Enhancer-Associated Genes in AML." *Nature*, vol. 526, no. 7572, 2015, p. 273.

Pennington, Jeffrey, et al. *Glove: Global Vectors for Word Representation*. 2014, pp. 1532–43.

Pham, Tuyen, et al. "Multi-National Banknote Classification Based on Visible-Light Line Sensor and Convolutional Neural Network." *Sensors*, vol. 17, no. 7, 2017, p. 1595.

Pigou, Lionel, et al. "Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video." *International Journal of Computer Vision*, vol. 126, no. 2–4, 2018, pp. 430–39.

Pongor, Lőrinc, et al. "A Genome-Wide Approach to Link Genotype to Clinical Outcome by Utilizing next Generation Sequencing and Gene Chip Data of 6,697 Breast Cancer Patients." *Genome Medicine*, vol. 7, no. 1, 2015, p. 104.

Qian, Ning, and Terrence J. Sejnowski. "Predicting the Secondary Structure of Globular Proteins Using Neural Network Models." *Journal of Molecular Biology*, vol. 202, no. 4, 1988, pp. 865–84.

Qian, Yanmin, et al. “Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, 2016, pp. 2263–76.

Raissi, Maziar. “Forward-Backward Stochastic Neural Networks: Deep Learning of High-Dimensional Partial Differential Equations.” *ArXiv Preprint ArXiv:1804.07010*, 2018.

Rajpurkar, Pranav, et al. “Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks.” *ArXiv Preprint ArXiv:1707.01836*, 2017.

Real, Pedro J., et al. “γ-Secretase Inhibitors Reverse Glucocorticoid Resistance in T Cell Acute Lymphoblastic Leukemia.” *Nature Medicine*, vol. 15, Dec. 2008, p. 50.

Reddi, Sashank J., et al. “On the Convergence of Adam and Beyond.” *ArXiv Preprint ArXiv:1904.09237*, 2019.

Reddy, Trips. *How Chatbots Can Help Reduce Customer Service Costs by 30%*. 17 Oct. 2017, https://www.ibm.com/blogs/watson/2017/10/how-chatbots-reduce-customer-service-costs-by-30-percent/.

Reiner, Anat, et al. “Identifying Differentially Expressed Genes Using False Discovery Rate Controlling Procedures.” *Bioinformatics*, vol. 19, no. 3, 2003, pp. 368–75.

Rhodes, Daniel R., and Arul M. Chinnaiyan. “Integrative Analysis of the Cancer Transcriptome.” *Nature Genetics*, vol. 37, no. 6s, 2005, p. S31.

Richardson, Alan. “Seismic Full-Waveform Inversion Using Deep Learning Tools and Techniques.” *ArXiv Preprint ArXiv:1801.07232*, 2018.

Robson, Mark E., et al. “American Society of Clinical Oncology Policy Statement Update: Genetic and Genomic Testing for Cancer Susceptibility.” *Journal of Clinical Oncology*, vol. 33, no. 31, 2015, pp. 3660–67.

Rong, Xin. “Word2vec Parameter Learning Explained.” *ArXiv Preprint ArXiv:1411.2738*, 2014.

Rosenberg, Philip S., et al. “Cancer Incidence in Persons with Fanconi Anemia.” *Blood*, vol. 101, no. 3, 2003, pp. 822–26.

Rost, Burkhard, et al. “Transmembrane Helices Predicted at 95% Accuracy.” *Protein Science*, vol. 4, no. 3, 1995, pp. 521–33.

Rost, Burkhard, and Chris Sander. “Prediction of Protein Secondary Structure at Better than 70% Accuracy.” *Journal of Molecular Biology*, vol. 232, no. 2, 1993, pp. 584–99.

Rumelhart, David E. “Parallel Distributed Processing: Explorations in the Microstructure of Cognition.” *Learning Internal Representations by Error Propagation*, vol. 1, 1986, pp. 318–62.

Salehinejad, Hojjat, et al. *Generalization of Deep Neural Networks for Chest Pathology Classification in X-Rays Using Generative Adversarial Networks*. IEEE, 2018, pp. 990–94.

Scalero, Robert S., and Nazif Tepedelenlioglu. "A Fast New Algorithm for Training Feedforward Neural Networks." *IEEE Transactions on Signal Processing*, vol. 40, no. 1, 1992, pp. 202–10.

Schessl, Joachim, et al. "Clinical, Histological and Genetic Characterization of Reducing Body Myopathy Caused by Mutations in FHL1." *Brain*, vol. 132, no. 2, 2008, pp. 452–64.

Schlingemann, Joerg, et al. "Profile of Gene Expression Induced by the Tumour Promotor TPA in Murine Epithelial Cells: Expression Profiling in Skin Carcinogenesis." *International Journal of Cancer*, vol. 104, no. 6, May 2003, pp. 699–708. *Crossref*, doi:10.1002/ijc.11008.

Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional Recurrent Neural Networks." *IEEE Transactions on Signal Processing*, vol. 45, no. 11, 1997, pp. 2673–81.

Shannon, Paul, et al. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks." *Genome Research*, vol. 13, no. 11, 2003, pp. 2498–504.

Shimamura, Akiko, et al. "Rsk1 Mediates a MEK–MAP Kinase Cell Survival Signal." *Current Biology*, vol. 10, no. 3, 2000, pp. 127–35.

Shin, Hoo-Chang, et al. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning." *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, 2016, pp. 1285–98.

Shiozawa, Yusuke, et al. "Gene Expression and Risk of Leukemic Transformation in Myelodysplasia." *Blood*, vol. 130, no. 24, 2017, pp. 2642–53.

Shurin, Michael R., et al. "FLT3: Receptor and Ligand. Biology and Potential Clinical Application." *Cytokine & Growth Factor Reviews*, vol. 9, no. 1, 1998, pp. 37–48.

Singh, Rajat, et al. "Synergistic Antitumor Activity of Anti-CD25 Recombinant Immunotoxin LMB-2 with Chemotherapy." *Clinical Cancer Research*, vol. 18, no. 1, 2012, pp. 152–60.

Smith, David A., et al. *Infectious Texts: Modeling Text Reuse in Nineteenth-Century Newspapers*. IEEE, 2013, pp. 86–94.

Smith, Leslie N. *Cyclical Learning Rates for Training Neural Networks*. IEEE, 2017, pp. 464–72.

Smith, Matthew L., et al. “Development of a Human Acute Myeloid Leukaemia Screening Panel and Consequent Identification of Novel Gene Mutation in FLT3 and CCND3.” *British Journal of Haematology*, vol. 128, no. 3, 2005, pp. 318–23.

Smith, Temple F., and Michael S. Waterman. “Identification of Common Molecular Subsequences.” *Journal of Molecular Biology*, vol. 147, no. 1, 1981, pp. 195–97.

Smyth, Gordon K. “Limma: Linear Models for Microarray Data.” *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Springer, 2005, pp. 397–420.

Spangler, Scott, et al. *Automated Hypothesis Generation Based on Mining Scientific Literature*. ACM, 2014, pp. 1877–86.

Spanhol, Fabio Alexandre, et al. *Breast Cancer Histopathological Image Classification Using Convolutional Neural Networks*. IEEE, 2016, pp. 2560–67.

Specht, Donald F. “A General Regression Neural Network.” *IEEE Transactions on Neural Networks*, vol. 2, no. 6, 1991, pp. 568–76.

Srivastava, Shriansh, et al. “Optical Character Recognition on Bank Cheques Using 2D Convolution Neural Network.” *Applications of Artificial Intelligence Techniques in Engineering*, Springer, 2019, pp. 589–96.

Steensma, David P., and Ayalew Tefferi. “The Myelodysplastic Syndrome (s): A Perspective and Review Highlighting Current Controversies.” *Leukemia Research*, vol. 27, no. 2, 2003, pp. 95–120.

Stone, Gregory O. “An Analysis of the Delta Rule and the Learning of Statistical Associations.” *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, 1986, pp. 444–59.

Tabatabai, Ghazaleh, and Michael Weller. “Glioblastoma Stem Cells.” *Cell and Tissue Research*, vol. 343, no. 3, 2011, pp. 459–65.

Takahashi-Yanaga, Fumi, and Toshiyuki Sasaguri. “The Wnt/β-Catenin Signaling Pathway as a Target in Drug Discovery.” *Journal of Pharmacological Sciences*, vol. 104, no. 4, 2007, pp. 293–302.

Testa, Ugo. “Leukemia Stem Cells.” *Annals of Hematology*, vol. 90, no. 3, 2011, pp. 245–71.

Thomas, James, and Jeff Brunton. *EPPI-Reviewer: Software for Research Synthesis*. 2007.

Thompson, Neil, and John Lyons. “Recent Progress in Targeting the Raf/MEK/ERK Pathway with Inhibitors in Cancer Drug Discovery.” *Current Opinion in Pharmacology*, vol. 5, no. 4, 2005, pp. 350–56.

Tiacci, Enrico, et al. “High-Risk Clonal Hematopoiesis as the Origin of AITL and NPM1-Mutated AML.” *New England Journal of Medicine*, vol. 379, no. 10, 2018, pp. 981–84.

Tibshirani, Robert, et al. “Diagnosis of Multiple Cancer Types by Shrunken Centroids of Gene Expression.” *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, 2002, pp. 6567–72.

Tominaga, Kouji, et al. “IL-12 Synergizes with IL-18 or IL-1β for IFN-γ Production from Human T Cells.” *International Immunology*, vol. 12, no. 2, Feb. 2000, pp. 151–60. *Crossref*, doi:10.1093/intimm/12.2.151.

Treangen, Todd J., and Steven L. Salzberg. “Repetitive DNA and Next-Generation Sequencing: Computational Challenges and Solutions.” *Nature Reviews Genetics*, vol. 13, no. 1, 2012, p. 36.

Tsutsumi, Yasuo, et al. “Site-Specific Chemical Modification with Polyethylene Glycol of Recombinant Immunotoxin Anti-Tac (Fv)-PE38 (LMB-2) Improves Antitumor Activity and Reduces Animal Toxicity and Immunogenicity.” *Proceedings of the National Academy of Sciences*, vol. 97, no. 15, 2000, pp. 8548–53.

Unger, Russ, and Carolyn Chandler. *A Project Guide to UX Design: For User Experience Designers in the Field or in the Making*. New Riders, 2012.

Valk, Peter J. M., et al. “Prognostically Useful Gene-Expression Profiles in Acute Myeloid Leukemia.” *New England Journal of Medicine*, vol. 350, no. 16, Apr. 2004, pp. 1617–28. *Crossref*, doi:10.1056/NEJMoa040465.

Vardiman, James W., et al. “The 2008 Revision of the World Health Organization (WHO) Classification of Myeloid Neoplasms and Acute Leukemia: Rationale and Important Changes.” *Blood*, vol. 114, no. 5, 2009, pp. 937–51.

Verhaak, R. G. W. “Mutations in Nucleophosmin (NPM1) in Acute Myeloid Leukemia (AML): Association with Other Gene Abnormalities and Previously Established Gene Expression Signatures and Their Favorable Prognostic Significance.” *Blood*, vol. 106, no. 12, Dec. 2005, pp. 3747–54. *Crossref*, doi:10.1182/blood-2005-05-2168.

Viereck, Ulrich, et al. “Learning a Visuomotor Controller for Real World Robotic Grasping Using Simulated Depth Images.” *ArXiv Preprint ArXiv:1706.04652*, 2017.

Wang, Lipo, et al. “Accurate Cancer Classification Using Expressions of Very Few Genes.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 4, no. 1, 2007, pp. 40–53.

Waterman, Michael S., and Mark Eggert. “A New Algorithm for Best Subsequence Alignments with Application to TRNA-RRNA Comparisons.” *Journal of Molecular Biology*, vol. 197, no. 4, 1987, pp. 723–28.

Waterman, Michael S., and Temple F. Smith. "Rapid Dynamic Programming Algorithms for RNA Secondary Structure." *Advances in Applied Mathematics*, vol. 7, no. 4, 1986, pp. 455–64.

Wei, Jun S., et al. "Prediction of Clinical Outcome Using Gene Expression Profiling and Artificial Neural Networks for Patients with Neuroblastoma." *Cancer Research*, vol. 64, no. 19, 2004, pp. 6883–91.

Weng, John J., et al. *Learning Recognition and Segmentation of 3-d Objects from 2-d Images*. IEEE, 1993, pp. 121–28.

Weng, Juyang, et al. *Cresceptron: A Self-Organizing Neural Network Which Grows Adaptively*. Vol. 1, IEEE, 1992, pp. 576–81.

Werbos, Paul. "Beyond Regression:" New Tools for Prediction and Analysis in the Behavioral Sciences." *Ph. D. Dissertation, Harvard University*, 1974.

Windpassinger, Christian, et al. "An X-Linked Myopathy with Postural Muscle Atrophy and Generalized Hypertrophy, Termed XMPMA, Is Caused by Mutations in FHL1." *The American Journal of Human Genetics*, vol. 82, no. 1, 2008, pp. 88–99.

Wingate, Andrew D., et al. "Nur77 Is Phosphorylated in Cells by RSK in Response to Mitogenic Stimulation." *Biochemical Journal*, vol. 393, no. 3, 2006, pp. 715–24.

Wishart, David S., et al. "DrugBank: A Comprehensive Resource for in Silico Drug Discovery and Exploration." *Nucleic Acids Research*, vol. 34, no. suppl_1, 2006, pp. D668–72.

Xiao, Han, et al. "Fashion-Mnist: A Novel Image Dataset for Benchmarking Machine Learning Algorithms." *ArXiv Preprint ArXiv:1708.07747*, 2017.

Xu, Lan, et al. "Genomic Landscape of CD34+ Hematopoietic Cells in Myelodysplastic Syndrome and Gene Mutation Profiles as Prognostic Markers." *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, 2014, pp. 8589–94.

Yin, Shihao, et al. *Traffic Sign Recognition Based on Deep Convolutional Neural Network*. Springer, 2017, pp. 685–95.

Yong, Ed. "A Waste of 1,000 Research Papers." *The Atlantic*, 17 May 2019, https://www.theatlantic.com/science/archive/2019/05/waste-1000-studies/589684/?utm_source=pocket-newtab.

Young, Andrew L., et al. "Clonal Haematopoiesis Harbouring AML-Associated Mutations Is Ubiquitous in Healthy Adults." *Nature Communications*, vol. 7, 2016, p. 12484.

Zanier, Romina, et al. "Fanconi Anemia C Gene Product Regulates Expression of Genes Involved in Differentiation and Inflammation." *Oncogene*, vol. 23, no. 29, June 2004, pp. 5004–13, doi:10.1038/sj.onc.1207677.

Zarrinkar, Patrick P., et al. “AC220 Is a Uniquely Potent and Selective Inhibitor of FLT3 for the Treatment of Acute Myeloid Leukemia (AML).” *Blood*, vol. 114, no. 14, 2009, pp. 2984–92.

Zeiler, Matthew D. “ADADELTA: An Adaptive Learning Rate Method.” *ArXiv Preprint ArXiv:1212.5701*, 2012.

---. *On Rectified Linear Units for Speech Processing*. IEEE, 2013, pp. 3517–21.

Zeng, Daojian, et al. *Relation Classification via Convolutional Deep Neural Network*. 2014.

Zhang, Dingxiao, et al. “Stem Cell and Neurogenic Gene-Expression Profiles Link Prostate Basal Cells to Aggressive Prostate Cancer.” *Nature Communications*, vol. 7, 2016, p. 10798.

Zhang, ML, et al. “Isolation and Sequencing of the CDNA Encoding the 75-KD Human Sperm Protein Related to Infertility.” *Chinese Medical Journal*, vol. 105, no. 12, 1992, pp. 998–1003.

Zhang, Zhilu, and Mert Sabuncu. *Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels*. 2018, pp. 8778–88.

Zheng, Yue, et al. “Gene Expression Patterns of Hippocampus and Cerebral Cortex of Senescence-Accelerated Mouse Treated with Huang-Lian-Jie-Du Decoction.” *Neuroscience Letters*, vol. 439, no. 2, 2008, pp. 119–24.

Zhong, Zhun, et al. “Random Erasing Data Augmentation.” *ArXiv Preprint ArXiv:1708.04896*, 2017.

Zipfel, Peter F., and Christine Skerka. “FHL-1/Reconectin: A Human Complement and Immune Regulator with Cell-Adhesive Function.” *Immunology Today*, vol. 20, no. 3, 1999, pp. 135–40.

Appendix 1: List of deep learning hyperparameters and defaults

β_i^T	Time-corrected bias on moment estimate of order *i* (Adam/Nadam)	0.9/0.999
β_i	Bias correction on moment estimate of order *i* (Adam/Nadam)	0.9/0.999
B	Batch size	4
η	Learning rate	0.001
ε	Epsilon (denominator stabilizer)	10^{-8}
μ	Signal gain	1.0
m	Momentum (in SGD + M)	0.9
ψ	Gradient noise	0.5
ρ	Gradient clipping	1.0
p	Dropout probability (per layer)	0.0
θ	Backpropagation loss threshold	0.05
γ	Weight decay	0.01
ζ	Cost-inertia decay	1.0

Appendix 2: Summary of deep learning results

A2.1: Digit-MNIST

MLP: 784, 128 (Range Sigmoid, p = 0.15), 10 (Softmax); $\eta = 0.01$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.001$, $\rho = \pm 1$, B = 4; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	98.5283	97.05
2	98.5133	97.04
3	98.51	97.02
4	98.4833	97.15
5	98.4783	97.18
6	98.5117	97.13
7	98.44	96.98
8	98.52	97.05
9	98.4717	97.14
10	98.5	97.14

Conv-net: 6@24x24 (ReLU; $\alpha = 0.15$), 864, 128 (Range sigmoid, p = 0.15), 10 (Softmax)

$\eta = 0.01$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.001$, $\rho = \pm 1$, B = 4; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	99.5883	98.2
2	99.91	98.34
3	99.725	98.3
4	99.2733	97.78
5	99.235	97.77
6	99.91	98.64
7	99.2783	97.79
8	99.685	98.28
9	99.2283	97.76
10	99.2483	98.07

A2.2: Fashion-MNIST

MLP: 784, 128 (Range Sigmoid, p = 0.1), 10 (Softmax); $\eta = 0.001$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.001$, $\rho = \pm 1$, B = 4; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	92.3467	88.47
2	92.3233	88.48
3	92.205	88.62
4	92.125	88.47
5	92.12	88.75
6	92.1183	88.75
7	92.0017	88.38
8	91.98	88.38
9	92.1167	88.59
10	91.9717	88.54

Conv-net: 6@24x24 (ReLU; $\alpha = 0.1$), 864, 128 (Range sigmoid, p = 0.1), 10 (Softmax); $\eta = 0.001$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.001$, $\rho = \pm 1$, B = 4; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	95.045	89.83
2	95.0883	89.9
3	93.5583	89.3
4	93.5667	89.3
5	93.8667	89.84
6	93.8633	89.82
7	93.83	89.5
8	93.835	89.49
9	94.3133	90.3
10	94.3067	90.26

A2.3: CIFAR-10

Conv-net: 32@30x30 (ReLU; $\alpha = 0.1$), 32@28x28 (ReLU, $\alpha = 0.1$), Max-pooling 2x2, 64@10x10 (ReLU, $\alpha = 0.3$), 64@8x8 (ReLU, $\alpha = 0.3$), 1024, 256 (Range sigmoid, $p = 0.05$), 128 (Range sigmoid), 10 (Softmax); $\eta = 0.001$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.001$, $\rho = \pm 1$, $B = 4$; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	70.36	60.88
2	69.912	60.75
3	70.762	59.83
4	71.306	59.71
5	71.256	60.21
6	71.404	60.39
7	71.032	60.38
8	71.282	60.51
9	73.734	62.49
10	72.872	61.98

A2.4: Text-to-function (T2F)

MLP: 1585, 256 (Range sigmoid), 196 (Range sigmoid), 10 (Softmax); $\eta = 0.001$, $\theta = 0.05$, $\psi = 0.5$, $\gamma = 0.01$, $\rho = \pm 1$, $B = 4$; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	99.6767	84.273
2	99.7845	84.273
3	99.7845	81.0089
4	99.7845	80.1187
5	99.6767	80.1187
6	99.569	81.6024
7	99.8922	79.2285
8	99.6767	81.0089
9	99.8922	81.0089

10	99.7845	81.8991
11	99.569	82.1958
12	99.7845	81.6024
13	99.7845	84.5697
14	99.569	84.8665
15	99.6767	82.4926
16	97.1983	77.7448
17	99.7845	83.0861
18	99.7845	81.8991
19	99.7845	81.8991
20	99.7845	81.8991
21	93.319	78.9318
22	96.5517	78.3383
23	98.2759	83.3828
24	97.9526	82.7893
25	99.2457	83.9763

Conv-net: 4@1x317 (ReLU, $\alpha = 0.1$), 1268, 256 (Range sigmoid), 196 (Range sigmoid), 10 (Softmax); $\eta = 0.001$, $\theta = 0.05$, $\psi = 0.5$, $\gamma = 0.01$, $\rho = \pm 1$, $B = 4$; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	99.6767	82.7893
2	99.7845	83.9763
3	99.8922	83.6795
4	99.8922	83.6795
5	99.1379	78.0415
6	98.9224	75.6677
7	99.7845	83.0861
8	99.7845	82.4926
9	99.3534	81.8991
10	99.4612	81.6024
11	99.7845	83.6795
12	99.7845	82.7893
13	99.4612	78.635
14	96.6595	77.7448
15	99.7845	84.273
16	99.569	83.6795
17	99.6767	81.8991

18	99.6767	80.7122
19	99.3534	81.8991
20	99.6767	83.9763
21	99.569	86.6469
22	99.7845	84.273
23	99.7845	84.273
24	99.8922	83.0861
25	99.8922	83.9763

A2.5: Context Recognition

Genetic recognition conv-net: 4@1x317 (ReLU, $\alpha = 0.15$), 1268, 256 (Range sigmoid), 196 (Range sigmoid), 10 (Softmax); $\eta = 0.001$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.01$, $\rho = \pm 1$, $B = 4$; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	94.8	93.9389
2	95.6714	94.35
3	96.9476	95.7278
4	95.6857	94.1167
5	95.569	94.0722

Therapeutic recognition conv-net: 4@1x317 (ReLU, $\alpha = 0.15$), 1268, 256 (Range sigmoid), 196 (Range sigmoid), 10 (Softmax); $\eta = 0.001$, $\theta = 0.01$, $\psi = 0.5$, $\gamma = 0.01$, $\rho = \pm 1$, $B = 4$; Nadam optimizer

Run	Train Accuracy (%)	Test Accuracy (%)
1	86.3333	81.4667
2	90.031	83.7222
3	86.8095	81.6167
4	86.0881	81.7056