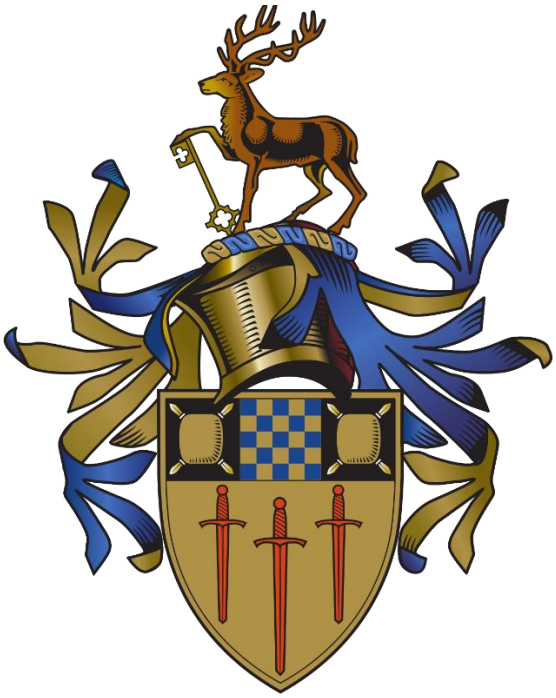
Use of Truth Discovery algorithms for improving the accuracy of single-cell RNAseq cell type identification



Simeon Stavrev

Faculty of Health and Medical Sciences

University of Surrey

This dissertation is submitted for the degree of

Biomedical Science (BSc) in 2022.

It is the original work of the author, and the work of others

is indicated by explicit references.

Word count: 5565

Acknowledgements:

Thank you to Dr. Alexessander Da Silva Couto Alves for his valuable guidance and support on this project as my supervisor. Thank you to Katie-Jo Miller for providing her own experimental scRNAseq cell type data, which was used in this project. Also, I want to extend my gratitude to my family, for their continued support throughout my degree.

Abstract

Correct identification of cells is essential to biological research and medicine. Single cell transcriptomic data is widely used to identify cells, but as it is generally complex and high-dimensional, most cell annotators inevitably include an amount of error in their results. Application of different cell annotators on the same single-cell RNAseq dataset often results in conflicting predictions for some of the cell samples and there aren't any standard solutions to this problem.

This project aims to investigate the feasibility of applying a machine learning technique called Truth Discovery to resolve conflicts in heterogeneous single-cell RNAseq cell type data. Truth Discovery literature was reviewed and considerations for application to cell type data were identified. Suitable iterative voting-based algorithms were selected and tested on synthetic mock cell type data. The algorithms achieved promising results and experimentation with the synthetic data methodology allowed for identifying favourable input data properties. The selected truth discovery algorithms were also applied on real experimental single-cell RNAseq predictions from three cell annotators to potentially investigate which cell annotator was the most reliable. The results showcased the potential of Truth Discovery for the proposed use case and prompted questions for further investigation.

Contents

1. Introduction and Background 6
1.1 Single-cell RNAseq 6
1.2 Cell type identification/annotation 7
1.3 Heterogeneity of data 8
1.4. Exploring a potential solution – Truth Discovery 8
1.5 Project Aims and Objectives 8
1.6 Truth Discovery and Machine Learning 9
2. Literature review 10
2.1 General Truth Discovery algorithm characteristics 11
2.2 Voting-based algorithms 13
2.3 Implemented Truth Discovery methodologies 14
2.4 Truth Discovery application in the real world 17
3. Methods and materials 18
3.1 Programming language and package used 18
3.2 Synthetic data experiment 18
3.2.1 Mock cell type dataset properties 19
3.2.2 Algorithm performance comparison 20
3.3 Real cell type data experiment 21
3.3.1 Preliminary experimental data 21
3.3.2 Processing of the experimental data 21
3.3.3 Kendall rank correlation of cell annotators with computed true labels 22
3.3.4 Trust scores of each cell annotator 22
3.4 Availability of the project's code implementation, materials used and obtained results 22
4. Results 23
4.1 Synthetic data experiment 23
4.1.1 1000x algorithm runs accuracy measurement 23
4.1.2 Decreased claim probability accuracy measurements 24
4.1.3 Increased number of sources (12) accuracy measurements 25
4.1.4. 100x100 algorithm runs accuracy measurement 26
4.2. Real data experiment 27
4.2.1 Kendall rank correlation of each annotator's predicted cell type labels with true labels computed by each Truth Discovery algorithm 27
4.2.2 Trust scores of each cell annotator given by each Truth Discovery algorithm 27

5. Discussion 28
5.1 Synthetic data experiment 28
5.2 Real data experiment 30
5.3 Conclusions 31
6. References 32

1. Introduction and Background

1.1 Single-cell RNAseq

Single-cell RNAseq(scRNAseq) is a contemporary technique in molecular biology which utilizes next generation sequencing and computational methods to qualify and quantify the transcriptomes of individual cells.(Tang et al., 2009)

Different scRNAseq experiments follow a similar basic strategy. First, single cells are captured and lysed. The abundant ribosomal RNA is often removed from samples via enzymatic degradation, then reverse transcription is performed to select for target mRNA, yielding cDNA. Subsequently, the obtained small amounts of cDNA are amplified by either PCR or in vitro transcription. Finally, the amplified cDNA is used for a sequencing library preparation. Sequences can be mapped using adaptors and RNA expression levels can be observed as gene reads per sample.(Kolodziejczyk et al., 2015) (Chu & Corey, 2012)

Normally, scRNAseq analysis includes hundreds of samples (different single cells) as opposed to standard bulk-RNAseq which typically includes only several samples (representing different homogeneous cell populations). The expression of thousands of different genes is observed for hundreds of different samples during scRNAseq. As a result, high-dimensional matrix datasets are generated which are a significant analytical challenge. (Stegle et al., 2015) (Z. Wang et al., 2009)

As the use of scRNAseq has gained traction, a wide variety of methodologies have been developed to analyse the resulting transcriptome datasets and try to tackle their many complications. (Ozsolak & Milos, 2011)

1.2 Cell type identification/annotation

Cell type identification is one of the most important questions in scRNAseq data analysis – it is often the first step or even the main goal of experiments. Cell type annotation is important not only for being able to correctly interpret transcriptome data. It can be applied across medicine and biology for a variety of uses. Knowledge of the cells' type in a sample is essential for investigating malignancies, immune responses and many other processes. (W. Ma et al., 2021)
Due to the complex nature of transcriptome data analysis, accurate cell type identification is still a challenge and there is an absence of a universally applicable methodology. (Z. Li & Feng, 2022)

There are two types of analysis depending on the availability of previous knowledge on the scRNAseq cell samples.:

1) If the single cell samples only include a defined set of previously known cell types, methodologies which make use of existing scRNAseq databases can be used. They leverage available reference cell type datasets to annotate the newly acquired samples by mapping them to the reference data. Due to their dependency on prior knowledge, the analytical algorithms which implement this approach are referred to as "Supervised". The outputs of three such cell annotators have been used to achieve the aims of this project – namely SingleR(Aran et al., 2019), Azimuth(Y. Hao et al., 2021) and SingleCellNet(Tan & Cahan, 2019).

2) In contrast, when cells must be identified "de novo", more complex "Unsupervised" analysis pipelines are required. Most often, they rely on dimensionality reduction and clustering of the data to identify the different cell types. Numerous variations exist, including some novel applications of Machine Learning techniques such as Neural networks and Meta-learning. (Peyvandipour et al., 2020) (Z. Li & Feng, 2022)(Kiselev et al., 2017)) (F. Ma & Pellegrini, 2020)

1.3 Heterogeneity of data

There are different methodologies which can be used to annotate the same scRNAseq dataset. Due to the complexity and variability of the scRNAseq cell typing process, different annotators often provide conflicting cell type labels for some samples.

1.4. Exploring a potential solution – Truth Discovery

While there might not be a one-suits-all cases methodology to address the problem of scRNAseq data heterogeneity, the results from multiple different cell type annotators could be accounted for at once, to improve the overall accuracy of the cell type identification process. The Truth Discovery machine learning technique implemented in this project aims to tackle such paradigms. It selects a true value (cell type) for an object (cell sample) from a pool of conflicting multi-source data (different cell annotators' labels) by computing each source's trustworthiness and each claim's belief score. It allows to effectively combine cell annotators' predictions, taking each of them into account, producing a single cell type label for each cell sample.

1.5 Project Aims and Objectives

Aims: This project aims to investigate the feasibility of applying the Truth Discovery technique to resolve conflicts in heterogeneous scRNAseq cell type data. It is meant provide an overview of relevant literature as well as identify limitations and guidelines for further research on applying the proposed novel approach to the cell typing problem.

Objectives:

- scRNAseq, Machine Learning and Truth Discovery literature will be reviewed to identify methodologies best suited for the properties of cell type datasets.
- An applicable code implementation of Truth Discovery algorithms will be selected and adapted for the research purpose.
- The algorithms' performance will be assessed using synthetic mock cell type data.
- The algorithms will be applied to experimental cell type predictions from three different scRNAseq cell annotator algorithms.

1.6 Truth Discovery and Machine Learning

Machine learning is a branch of Artificial Intelligence. It entails computational and analytical techniques which are used to draw inferences from patterns in multi-dimensional data. They are generally used to create predictive models and to obtain specific information or data characteristics. (Camacho et al., 2018) The data generated in biological and medical research is often complex and heterogeneous (poorly understood). Machine Learning techniques have been used to aid biological data analysis for several decades. (Stormo et al., 1982) (Larrañaga et al., 2006) With the advances in Next-generation sequencing technology, complex datasets have become increasingly more ubiquitous. As a result, there has been an explosive growth of interest in applying Machine Learning algorithms to biological research. The beforementioned scRNAseq cell type annotator methodologies are also examples of ML algorithms.

Truth Discovery is part of Machine Learning and encompasses a broad spectrum of algorithms which aim to resolve conflicts in heterogenous data by estimating the reliability of sources and their claims (by keeping track of **source trustworthiness** and **claim belief** scores). The main driver for the development of truth discovery techniques has been the ever-increasing amount of data available on the internet. There can often be conflicting information provided by different sources about the same objects.(Q. Li et al., 2014)

An illustrating example of this problem is a case where different websites(**sources**) claim different authors(**claims**) for the same book(**object**). Analogically, Truth Discovery algorithms can be applied to cell type data in the form: cell annotator(**source**) – cell sample (**object**) – cell type(**claim**).

2. Literature review

This Literature Review aims to aid the understanding of the Truth Discovery concept by characterizing the wide Truth Discovery field. Examples are provided and the reasoning behind selecting the algorithms used in this project is explained.

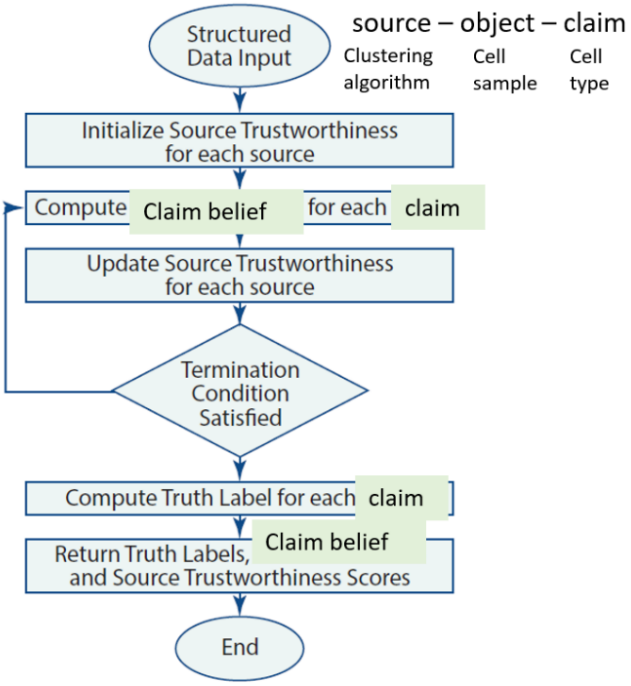
Glossary

Term	Definition
Claim	A claim is the predicted value of an object by a source
Claim belief	A score which quantifies the veracity of a claim
Ground truth	A **known** true value of an object
Label	A value of an object **predicted** by a source
Object	An entity which has a **single** true value
Source	A **source** makes claims about objects (e.g "annotator 1" claims cell 1 is a macrophage)
Source trustworthiness	A score which quantifies the reliability of a source, given the belief scores of its claims.

According to Berti-Équille & Borge-Holthoefer, there are three general approaches to inferring truth from a dataset with conflicting claims: Content-based, Recommendation-based and Evidence-based.

Recommendation- and Evidence- based approaches are designed to work as part of a more complex, dedicated system and to incorporate external information such as evidence or recommendations on sources (as their names imply).

Algorithms utilizing the **Content-based** approach aim to find true claims solely by iterating through the input data. They take a structured dataset(matrix) as input and iteratively compute and update source trustworthiness and claim belief scores. Results are generated either when the algorithm reaches a previously set iteration limit or when convergence has been achieved (scores have stopped changing significantly). (Figure 2.1)



(Berti-Équille and Borge-Holthoefer, 2015)

Figure 2.1

2.1 General Truth Discovery algorithm characteristics

Truth Discovery techniques can be characterized based on three dimensions: **method**, **data input** and **output results**. General summaries describing these important characteristics are organized in the tables below:

Table 2.1

Method features	Description
Initialization/Parameter setting	Some methods require a complex initialization process and parameterization, in which cases a ground truth dataset or other prior knowledge might be necessary (e.g. AccuSim by Dong et al., 2009 and LTM by Zhao et al., 2012)
Convergence and stopping criterion	Algorithms can have different termination conditions. Most often a fixed number of iterations are performed, or a convergence function is used, where iteration continues until the distance between successive claim belief scores becomes lower than a given threshold.

Complexity and scalability	There is always a trade-off between efficiency and scalability, method assumptions and accuracy. Thus, algorithms must be carefully selected and prepared according to the intended use. (X. Wang et al., 2015)
Supervised/Unsupervised	Some algorithms require fitting to a training dataset (with known ground truths) before analysing real data – **supervised**. Others can directly be used to analyse data with unknown ground truths – **unsupervised**.

Table 2.2

Data Input features	
Data type	Most algorithms can work with categorical, numerical, or string(text) input data values. MLE by D. Wang et al., 2012 requires pre-processing to convert the data into Boolean values (assign a True/False value for each claim)
Similarity of claim values	Algorithms such as TruthFinder by Yin et al., 2008 and AccuSim by Dong et al., 2009 recognise similar string/text values claimed for the same data item and consider them as mutually supportive in the truth computation process.

Table 2.3

Output features	
None, single or multiple true values	Only few methods can handle multiple true values for an object – LTM by Zhao et al., 2012 and MLE by D. Wang et al., 2012. The opposite is also rare - few methods can be used if none of the claimed values are true.(Zhi et al.,2015)

2.2 Voting-based algorithms

Content-based algorithms can be divided further according to their specific truth computation techniques. While some are voting-based and rely on "counting" the number of agreeing/disagreeing sources for each data item, there are also other probabilistic models which have more complex truth computation which uses Bayesian inference, Gibbs sampling and matrix diagonalization. Voting-based truth discovery algorithms were selected as the most suitable for this project due to their properties. They offer scalability and have an unsupervised mode of work. (Berti-Équille & Borge-Holthoefer, 2015)Thus, they are more relevant to the problem of cell type identification, as ground truths on cell types are seldom available, especially in research.

In *Table 2.2*, Voting-based algorithms are compared to the probabilistic LTM and MLE to illustrate key differences.

Table 2.4.		Truth Discovery algorithms		
Dimension	Subdimension	Voting-based algorithms (TruthFinder, Sums etc.)	Latent Truth Model (LTM)	Maximum Likelihood Estimation (MLE)
Method	Complex parameter setting	No	Yes	No
	Convergence	Yes	No	No
	Scalability	Yes	Yes	No
	Prior knowledge on sources/dataset required	No	Yes	No
Input data	**S**tring(text), **N**umerical, **C**ategorical, **B**oolean	S/C/N	S/C/N	B
	Claim similarity consideration	Yes	No	No
Output	Single/Multiple truths	Single truth only	Multiple truths	Single truth only

2.3 Implemented Truth Discovery methodologies

The baseline voting algorithm is **MajorityVoting**, it is a simplistic yet reliable approach for elimination of conflicts among multi-source data where an object is assigned the most "voted" for label among sources. (Figure 2.2.)

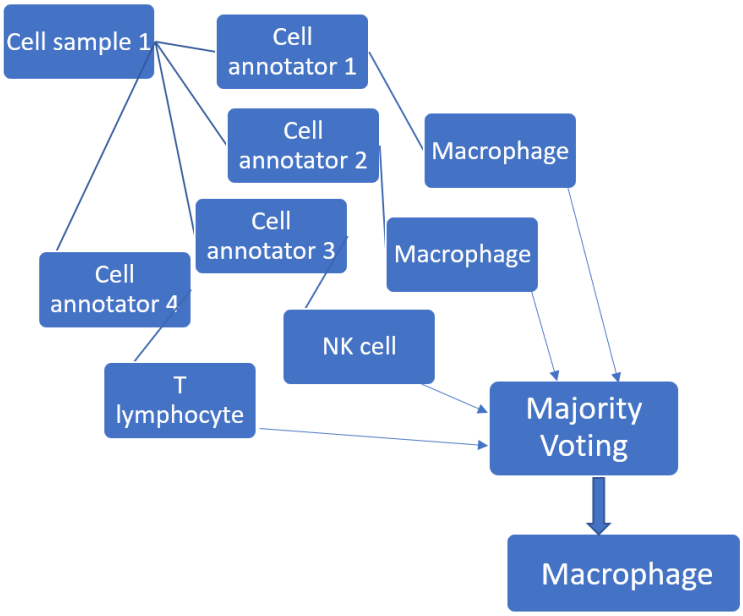


Figure 2.2:

A Majority voting example where four sources (cell annotators) make claims about the label of the object – "Cell sample 1". The selected true label is Macrophage as it has the highest number of "votes".

This simple voting methodology assumes classifiers are equally trustworthy, which brings about its limitations. MajorityVoting can be misleading in the case of a tie (equal number of votes for two claims) or when all sources aren't equally reliable (e.g. some consistently make more accurate predictions than others)

Iterative voting-based algorithms build on the intuitive voting logic but also aim to tackle its limitations. By using iterative computation formulas, they keep track of source trustworthiness and claim belief scores which are inferred solely from the input data. Thus, the accuracy of aggregated results can be improved by giving each source a different weight in the final "vote" for each object. Each algorithm is designed differently according to its purpose, with varying advantages and limitations for different data scenarios. Described below are the main ideas of the iterative voting algorithms implemented in this project, along with their computation formulas.

Symbol annotations used in algorithms' computational formulas:

The same **symbol** annotations are used for the formulas of the three truth discovery algorithms. We have a set of sources **S** each asserting a set of claims **Cs** where **s** designates the source. Iterative truth discovery algorithms iterate to calculate the trustworthiness (**T**) of each source $\mathbf{T^i(s)}$ at iteration **i** in terms of the belief in its claims in the previous iteration $\mathbf{B^{i-1}(Cs)}$, and belief in each claim $\mathbf{B^i(c)}$ in terms of $\mathbf{T^i(Sc)}$, where **Sc** is the set of all sources asserting the claim **c**. Iteration continues until a predefined limit or convergence(when trustworthiness and belief scores stop changing significantly).

Investment (Pasternack & Roth, 2010)

Sources are viewed as investors and "invest" their trustworthiness uniformly across their claims. After each iteration sources get proportional trustworthiness "returns" for each claim they invested in (i.e. the source which invested most in a claim receives the most return from that claim). Source trustworthiness is calculated as the sum of the beliefs in the source's claims, weighted by the proportion of trust previously contributed to each (relative to the other investors). Belief score of claims grows according to a non-linear function $G(x) = x^g$, defined on the sum of sources' investments in their claims where g=1.2.

$$T^i(s) = \sum_{c \in C_s} B^{i-1}(c) \cdot \frac{T^{i-1}(s)}{|C_s| \cdot \sum_{r \in S_c} \frac{T^{i-1}(r)}{|C_r|}}$$

$$B^i(c) = \mathcal{G}\left(\sum_{s \in S_c} \frac{T^i(s)}{|C_s|} \right)$$

TruthFinder (Yin et al., 2008)

TruthFinder is a pseudoprobabilistic algorithm – it calculates the "probability" of a claim to be true by assuming that each source's trustworthiness is the probability of it being correct. Each source's trustworthiness is the average of its claims' belief scores. The computation of claim beliefs includes optional parameters. **ρ** is an influence parameter between zero and one it controls the influence which claims can have on each other - if two claims from different sources are similar – e.g. lymphocyte and T lymphocyte, their belief score is additionally increased according to **ρ**. **γ** is a dampening factor which aims to account for cases when sources aren't completely independent and there is a possibility that some copy others' claims. (Theoretically, this property could be useful for cell typing data if some clustering algorithms utilize a similar method to annotate cells and thus make similar errors.) The formula below is for the basic version of TruthFinder - the optional parameters are omitted for the sake of simplicity, as changing them didn't provide any value for the project and they were kept at their default values.

$$T^i(s) = \frac{\sum_{c \in C_s} B^{i-1}(c)}{|C_s|}$$

$$B^i(c) = 1 - \prod_{s \in S_c} \left(1 - T^i(s)\right)$$

Sums (Pasternack & Roth, 2010)

Sums is based on the Hubs and Authorities algorithm by Kleinberg, 1999. Each source's trustworthiness is equal to the sum of the sources making the same claim. Each claim's belief score is equal to the sum of trustworthiness of the sources making the claim. In other words, the more a source's claims are linked to other sources' claims the more trustworthy the source is deemed.

$$T^i(s) = \sum_{c \in C_s} B^{i-1}(c) \quad B^i(c) = \sum_{s \in S_c} T^i(s)$$

2.4 Truth Discovery application in the real world

There are many examples of successfully implemented Truth Discovery methods to tackle real-world problems. Data fusion is a form of truth discovery which is widely used in academia and industry nowadays. (Bleiholder & Naumann, 2009) A probabilistic truth discovery methodology has also been applied in online health communities for evaluating reliability of patient reviews and discovering rare drug side effects. (Mukherjee et al., 2014)

The high-performance algorithms for resolving conflicts in heterogeneous data selected for this project have proven to be effective when used with synthetic as well as real-world data in the literature, reaching up to 99% accuracy. (Pasternack & Roth, 2010) (Yin et al., 2008)

There is enough evidence suggesting that Truth Discovery algorithms can be successfully applied to tackle the contemporary problem of resolving conflicts in heterogeneous scRNAseq cell typing datasets.

3. Methods and materials

3.1 Programming language and package used

Truth Discovery and ML algorithms in general can be implemented by using different programming languages. The Python programming language was used for this project as it is the most widely used for ML methods. (J. Hao & Ho, 2019)

A wide variety of available Truth Discovery algorithms on the web (uploaded on GitHub - an open-source code sharing platform) were reviewed and tested to select ones which are up-to-date and applicable for the research purpose. Issues were encountered with some of the available algorithms – mainly due to many of them being outdated or impractical. A significant amount of time was required to filter out non-feasible options. The search was narrowed down to a python package called "truthdiscovery"(Singleton, 2019). It implements exemplar voting-based truthdiscovery algorithms and provides an appropriate methodology for custom truthdiscovery data synthesis (used to create the mock cell type dataset). Being able to run the package successfully has been essential for the conduction of this project.

The MajorityVoting, TruthFinder, Investment and Sums algorithms were used in this project as each works with slightly different assumptions and methodology. Two general experiments were carried out – one on synthetic data and one with a real cell type dataset.

3.2 Synthetic data experiment

The available data synthesis methodology for truth discovery experimentation was used from the "truthdiscovery" python package. Mock cell type datasets were used to investigate and compare the performance of the selected algorithms. The experiment was set up so that each time an algorithm is run, a synthetic dataset is generated for it, where the probability of each source's claims to be correct is randomly drawn from a uniform distribution (0.0-1.0) using the function "numpy.random.uniform" of the NumPy python package(Harris et al., 2020). Ground truths are also provided with each dataset to allow measurement of each algorithm's accuracy with the mock cell type data.

3.2.1 Mock cell type dataset properties

The following **4**x**100**x**20** format was used to create a matrix resembling a real scRNA-seq cell typing dataset - **4** sources (cell annotators) provide claims (cell type labels) for **100** objects (cell samples) choosing from **20** possible cell type labels. Table 3.2 is an illustrative example (the actual implementation uses numbers instead of text/cell type labels). As opposed to this example, real cell typing datasets can sometimes come with less than 100% completeness of values. There are cases when an annotator(source) can't classify a cell sample(object), then the said source doesn't provide a claim for the said object. The data synthesis methodology of the "truthdiscovery" library accounts for this by incorporating a **claim probability** parameter (set to 100% by default), which can be changed to for the sake of experimentation. (i.e. If claim probability is set to 50%, each source in the generated dataset will provide claims only for a randomly selected 50% of the objects)

Table 3.2: Illustrative synthetic dataset

Example matrix	**Cell sample 1**	**Cell sample 2**	**Cell sample 3**	**Cell sample…**
Cell annotator 1	Macrophage	B lymphocyte	NK cell	…
Cell annotator 2	Neutrophil	B lymphocyte	Dendritic cell	…
Cell annotator 3	Macrophage	B lymphocyte	T lymphocyte	…
Cell annotator 4	Macrophage	B lymphocyte	B lymphocyte	…

3.2.2 Algorithm performance comparison

The performance of the selected truth discovery algorithms was measured by comparing the computed true labels with the provided ground truth labels for each synthetic object. Accuracy scores were collected as results for each algorithm where accuracy is defined as the percentage of correct true labels computed by the algorithm. The results were plotted in Boxplot graphs to enable a visual representation of the varying accuracy distribution.

3.2.2.1 1000x algorithm runs accuracy measurement, 100% claim probability, 4 sources

Each of the selected truth discovery algorithms was run 1000 times (each time on a newly generated dataset, as described above) on a dataset where 4 sources make claims for 100% of the objects.

3.2.2.2 Decreased claim probability accuracy measurements

Each of the selected truth discovery algorithms was run 1000 times (each time on a newly generated dataset, as described above) on a dataset where 4 sources make claims for 75% and 50% of the objects.

3.2.2.3 Increased sources accuracy measurements

Each of the selected truth discovery algorithms was run 1000 times (each time on a newly generated dataset, as described above) on a dataset where 12 sources make claims for 100% and 75% of the objects. This was done to investigate how algorithm performance is affected by the number of sources.

3.2.2.4 100x100 algorithm runs accuracy measurement, 100% claim probability, 4 sources

To test for the consistency and validity of the data synthesis methodology and the subsequent algorithm performance, 100 average accuracy scores of 100 runs each were also collected (10000 total) for each of the selected truth discovery algorithms.

3.3 Real cell type data experiment

3.3.1 Preliminary experimental data

The used experimental scRNAseq cell type data was provided by Katie-Jo Miller (a fellow University of Surrey alumna) and is the result of her own work as part of her research project - "Single Cell RNA Sequencing Analysis of Peripheral Blood Mononuclear Cell Samples from COVID-19 Patients to Find How Cell Type Proportions and Gene Expression Change with Severity", 2021. She used publicly available scRNA-seq datasets of COVID-19 patients from the Human Cell Atlas COVID-19 Registry. The scRNAseq transcriptome data was normalised and scaled. Principal Component Analysis (PCA) was applied to it to identify 20 principal components with the 3000 most variable genes. K Nearest Neighbours (KNN) was sequentially used to cluster the data along the identified principal components. Three different scRNAseq annotating algorithms were then tested on the data to find out the cellular identity of each cluster – **SingleR**(Aran et al., 2019), **Azimuth**(Y. Hao et al., 2021) and **SingleCellNet**(Tan & Cahan, 2019). These algorithms are of the supervised type which leverage existing reference cell type datasets to annotate their input data. Because different reference datasets are used by each, the three algorithms yielded slightly different results, inevitably incorporating an amount of error in identifying the cell type of each cluster. The resulting predictions(claims) of each annotator constitute a suitable candidate for testing the truth discovery technique.

3.3.2 Processing of the experimental data

Each algorithm's cell type predictions were processed before they were converted into "claims" format to use with the selected Truth Discovery algorithms. Because each of SingleR, SingleCellNet and Azimuth have separate methodologies and use a different reference dataset, each had different specificity for certain types of cells – e.g. SingleR can distinguish both CD4 and CD8 T cells while SCN would classify both of these cells as just "T cell". In order to apply Truth Discovery to the three sets of claims, the data was standardized by simplifying each cell type label to the most general level within the three datasets – e.g. cell type labels such as "CD4 T cell" or "CD 8 T cell" in the SingleR dataset were converted to "T cell". 1000 cell type labels (for the same cell samples) were extracted from each annotator dataset, processed using Microsoft Excel as described above and saved as ".csv" files to use with the Truth Discovery algorithms.

3.3.3 Kendall rank correlation of cell annotators with computed true labels

Due to the absence of ground truth knowledge about the real cell types of the cell samples, each annotator's correlation with the computed true labels was tested to experimentally investigate the reliability of each cell annotator. A python implementation of the **Kendall rank correlation** statistical method from the scikit-learn package(Pedregosa et al., 2011) was used to measure the said correlation.

3.3.4 Trust scores of each cell annotator

The selected Truth Discovery algorithms assign trust scores to each **source**. The trust score of a source is its probability of being correct (as perceived by the Truth Discovery algorithm giving the score). The resulting final trust scores for each cell annotator were collected to observe how each Truth Discovery methodology interacts with the cell annotator sources.

3.4 Availability of the project's code implementation, materials used and obtained results

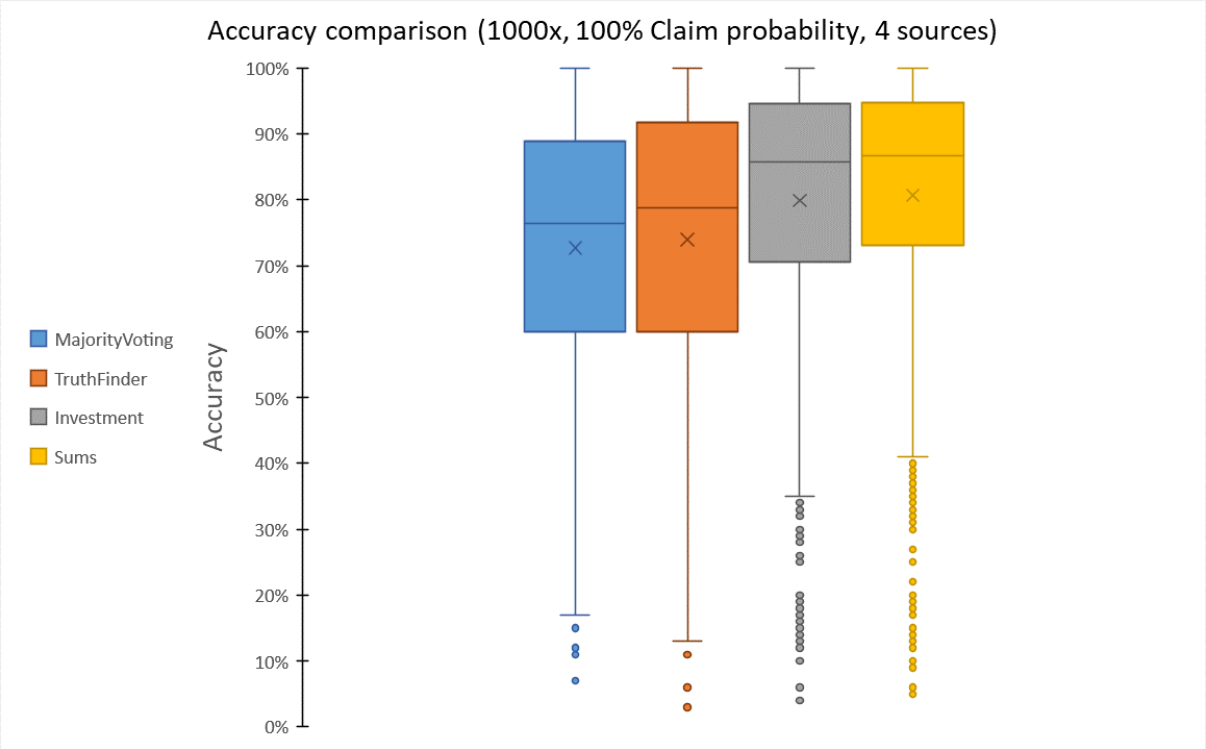
The files containing the python code implementation of both experiments, the used real cell type claims(anonymised) and all results can be viewed at the GitHub repository of the project: https://github.com/simeonstavrev/Truth-Discovery-application-to-heterogeneous-cell-type-data

4. Results

4.1 Synthetic data experiment (X designates **mean** values in all Boxplots graphs)

4.1.1 1000x algorithm runs accuracy measurement

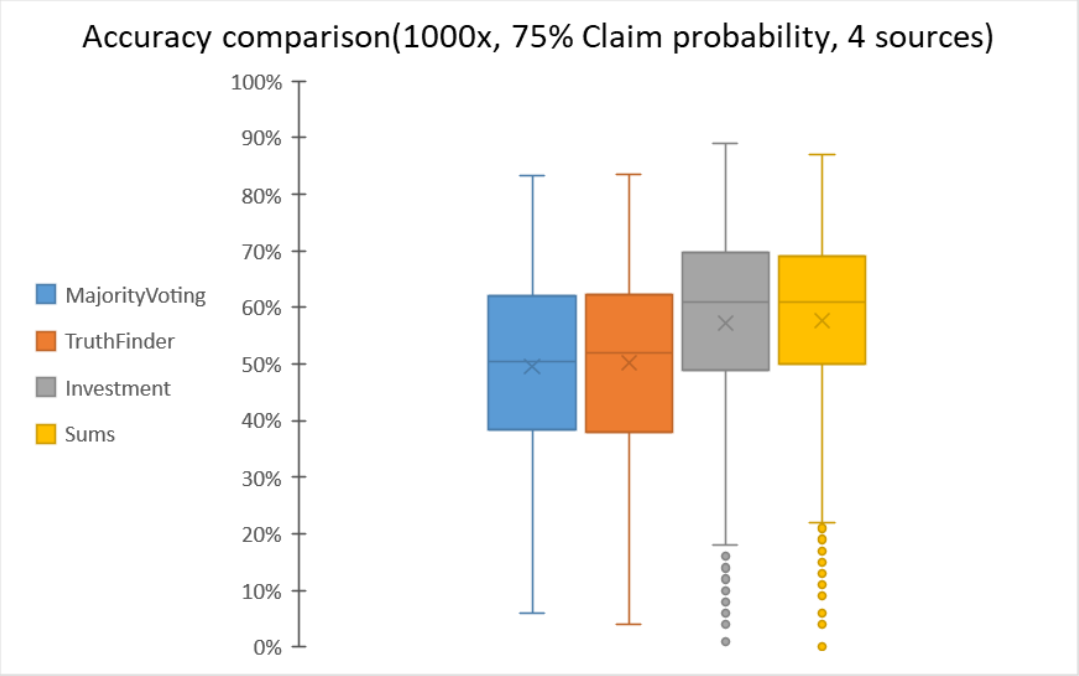
Graph 4.1



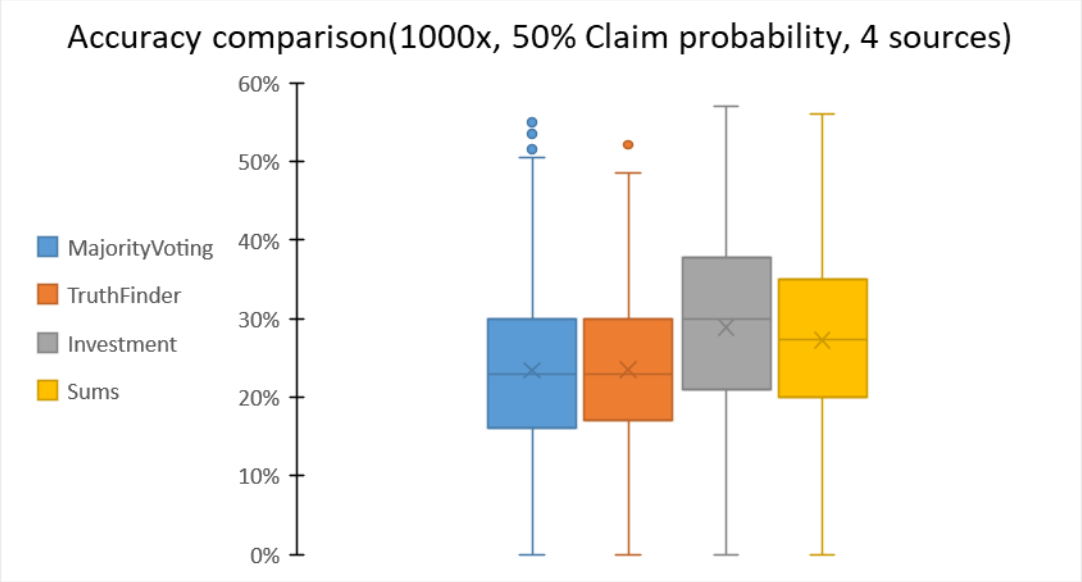
Graph 4.1 Findings: As observed in Graph 5.1, all four Truth Discovery algorithms had an average accuracy over 70%. MajorityVoting and TruthFinder had higher variance of results (ranging from around 10% to 100%). The Investment and Sums algorithms performed best with three quartiles (75%) of their results above 70% accuracy.

4.1.2 Decreased claim probability accuracy measurements

Graph 4.2.



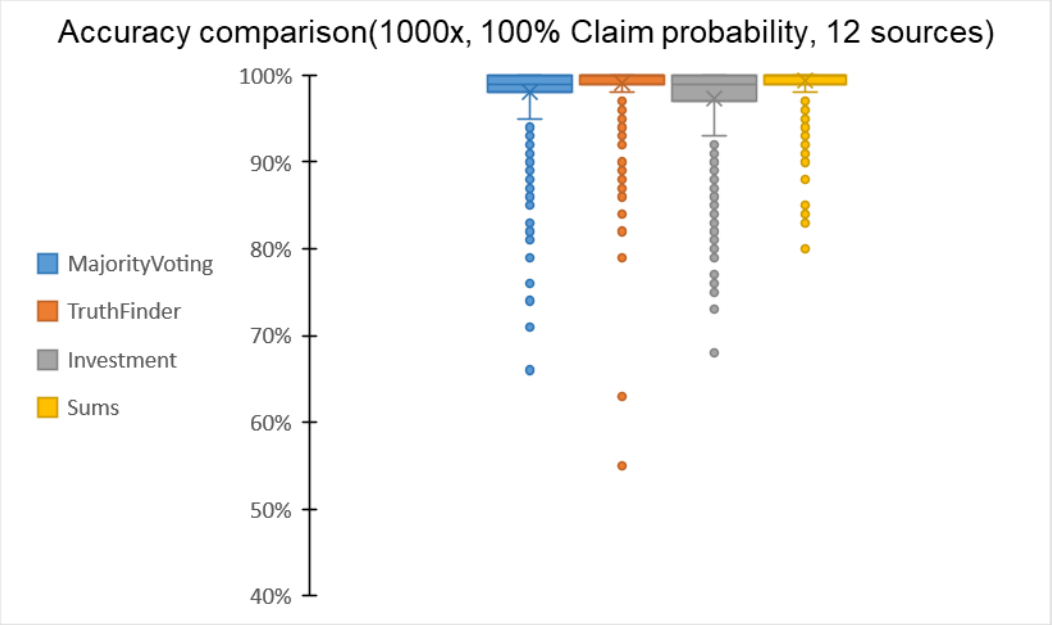
Graph 4.3



Graphs 4.2 and 4.3 Findings: All algorithms' average accuracy decreased to around 50-60% with a 25% decrease in claim probability, these scores were nearly halved with a further 25% decrease in claim probability. The Investment and Sums had a slight advantage over the other two.

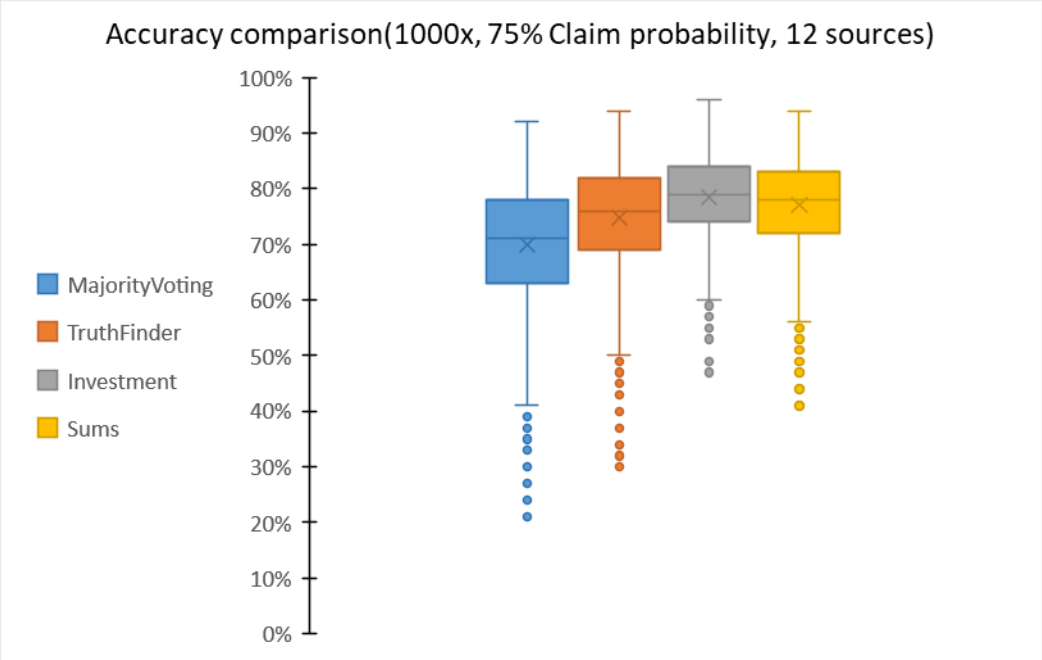
4.1.3 Increased number of sources (12) accuracy measurements

Graph 4.4



Graph 4.4 Findings: All four algorithms achieved extremely high scores (all 4 boxplot quartiles above 90%) and relatively low results variance when the source number was tripled.

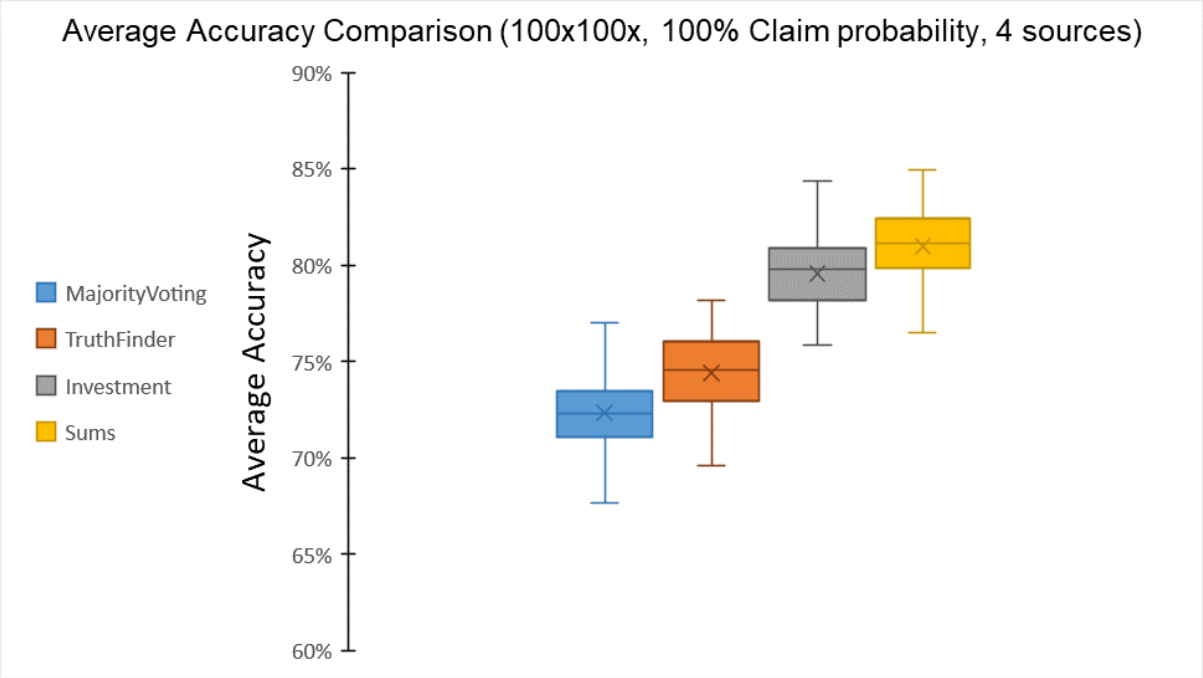
Graph 4.5



Graph 4.5 Findings: The algorithms achieved around 75% average accuracy, with relatively low variance of results, compared to Graph 4.2.

4.1.4. 100x100 algorithm runs accuracy measurement

Graph 4.6



Graph 4.6 Findings: Each result is an average of 100 runs. Mean values for each algorithm (10000 runs) are nearly identical to the mean values in Graph 4.1 (1000 runs). There aren't any outliers.

4.2. Real data experiment

4.2.1 Kendall rank correlation of each annotator's predicted cell type labels with true labels computed by each Truth Discovery algorithm

Table 4.2

	MajorityVoting	TruthFinder	Investment	Sums
SingleCellNet	0.4508	0.4962	0.4962	0.4962
azimuth	0.7594	0.7289	0.7289	0.7289
singleR	0.7401	1.0000	1.0000	1.0000

Table 4.2 Findings: singleR's labels had 100% correlation with the true labels computed by the Iterative voting algorithms. In contrast, SingleCellNet achieved the lowest correlation with them (0.4962) which was still higher than with MajorityVoting (0.4508). The Azimuth annotator achieved the highest correlation with MajorityVoting.

4.2.2 Trust scores of each cell annotator given by each Truth Discovery algorithm

Table 4.3

	MajorityVoting	TruthFinder	Investment	Sums
SingleCellNet	1.0000	0.5818	<0.0001	0.0025
azimuth	1.0000	0.6902	0.9975	0.9999
singleR	1.0000	0.6903	1.0000	1.0000

Table 4.3 Findings: singleR has been viewed as entirely trustworthy by the Investment and Sums algorithms with an extremely high trust score of 1.0 while SingleCellNet has almost been disregarded by the same algorithms (with trust scores <0.01). All trust scores given by MajorityVoting are equal to 1.0 as it isn't an iterative algorithm and implements simple voting without accounting for sources' trustworthiness – it takes all claims as equal.

5. Discussion

5.1 Synthetic data experiment

5.1.1 1000x algorithms accuracy comparison (*Graph 4.1)*

All algorithms' accuracy results presented with a relatively wide range of variance because data is generated entirely randomly. Especially MajorityVoting and TruthFinder were more sensitive to this - their results vary from ≈15% to 100% accuracy). On the other hand, the Investment and Sums algorithms had notably less variance (≈40% to 100%), suggesting that they are more consistent and reliable for handling complex data. They also computed true labels with a higher average accuracy (≈81%) compared to the other two. The baseline MajorityVoting showcased the usefulness of simple voting for resolving conflicting resolutions, by achieving an average accuracy of ≈ 72%. Unexpectedly, the TruthFinder didn't offer any advantage over simple voting on this synthetic data format. These results suggest that the Investment and Sums algorithms are better suited for application on cell type data and have the potential of achieving high accuracy. These algorithms both have similar method assumptions – the more a source's claims are connected to another source's claims, the more it is trusted. The TruthFinder has been designed to resolve conflicts in heterogeneous data provided by websites. It approaches the problem of source trustworthiness probabilistically, possibly due to the chances of human error when considering websites. As cell type annotators are in essence entirely computational, a more robust approach (such as the one behind Investment and Sums) to resolve conflicts in their claims might be more effective.

5.1.2 Decreased claim probability accuracy measurements (*Graph 4.2. and Graph 4.3)*

The claim probability of sources was decreased to test and speculate about the algorithms' ability to handle incomplete datasets, resembling real scenarios when some cell annotators aren't able to classify all cell samples. Accuracy of the algorithms sharply declined as claim probability was reduced. The results of this test suggest that the algorithms are highly sensitive to missing claims in a similar input data format. The observed steep drop in accuracy might be due to the relatively low number of sources in the synthetic experiment (only 4). At claim probability 50% for each source, the probability of having just one or two claims (which might not be true) for an object is much higher compared to if there are more sources to account for others not making a claim for a certain object.

5.1.3 Increased number of sources accuracy measurements (*Graph 4.4 and Graph 4.5)*

The results clearly indicate that the increased number of sources significantly improved algorithms' performance (all achieved ≈99% average accuracy for the 100% claim probability simulation). The suggestion made in the reduced claims discussion was confirmed– when sources were increased from originally 4 to 12, all algorithms scored ≈15-20% better average accuracy in 75% claim probability. The observed variance of results was also lower in both 100% and 75% claim probability, 12 sources, compared to the original 100% claim probability, 4 sources simulation. Overall, this experiment strongly suggests that more sources (cell annotators) in the input dataset are highly beneficial for achieving a more accurate result using the voting- based truth discovery methodology.

5.1.3 100x100 algorithm accuracy comparison (*Graph 4.6)*

The mean results of the 100x100 experiment are almost identical to the mean values of the primary 1000x runs test. The results confirm the consistency of the synthetic data methodology and the algorithms' performance. This suggests that the experimental methodology used for this project is reliable and further investigations with various parameter adjustments can be carried out. As expected, there were no outliers in the results of this experiment (Graph 4.6.) due to each score being an average of 100 accuracy scores.

Limitations and future steps

Although the Truth Discovery algorithms generally achieved high average accuracy scores in the 4 sources simulation, they all included outlier results of accuracy near 0%. Even if rare, the possibility of such low results must be eliminated before any practical application. The increased sources results suggest that in fact the selected algorithms are much more effective when the claims data includes more sources, which needs to be considered as a limitation for real data use where there aren't many sources available. Furthermore, the observed results might not transfer to real experimentation due to various possible specifics of datasets which aren't accounted for by the selected truth discovery algorithms. Voting-based iterative algorithms cannot account for similar source mistakes (e.g. if two cell annotators have similar methodologies and make the same mistakes). By default, their fundamental voting assumption relies on the majority of sources to be independent and at least over 50% of them to provide correct claims. (Berti-Équille & Borge-Holthoefer, 2015) The selected truth discovery algorithms also rely on the assumption that the true value of an object is provided among the claims, they don't account for cases where no correct claim is provided for an object. (which might be the case for cell type data, especially in "de novo" cell annotation)

5.2 Real data experiment (*Table 4.2 and Table 4.3)*

The real data experiment was carried out to investigate the feasibility of applying the Truth Discovery algorithms to experimental biological data. All three iterative voting algorithms (TruthFinder, Investment, Sums) generated “true" labels equal to SingleR’s claims (Kendall rank correlation of 1.0). It was given an extremely high trust score of 1.0 by the Investment and Sums algorithms, essentially suggesting that the probability of SingleR being correct is 100%. The Azimuth cell annotator also received high trust scores by the iterative algorithms and achieved 0.7289 correlation with the computed truths. SingleCellNet had the lowest correlation with the iterative voting true labels (0.4962) which was still higher compared to the correlation achieved with MajorityVoting (0.4508). This points towards the ability of the three iterative algorithms to account for and make use of all claims even if their source is deemed unreliable in general. The SingleCellNet claims dataset included a significant amount of empty claims (unrecognized cells) – around 17%, which might have contributed to it obtaining an extremely low trust score (≈ 0) with the Investment and Sums algorithms. Although each has a different computational method, the three iterative truth discovery algorithms achieved identical results with the cell type dataset – all claims made by SingleR were considered true. This strongly suggests that SingleR is the most suitable cell annotator (of the three tested ones) for use with the available data. The high accuracy and reliability of SingleR is also noted in an existing benchmark comparison of cell annotator algorithms where it is one of the best performing ones. (Abdelaal et al., 2019) On the other hand, SingleCellNet’s origin paper (Tan & Cahan, 2019)suggests the annotator is better suited for classification of engineered cells, which hasn’t been the case in the used dataset (cell samples are from covid patients). This might explain why it achieved lower trust scores and lower correlation with the computed true labels.

Limitations and future steps:

As observed in the synthetic data experiment, the selected voting- based algorithms don’t perform very well when there are only few sources and also unclassified samples, which is why the obtained correlations and trust scores must be taken with caution. This limitation comes around because of the underlying majority voting assumption, as mentioned in the limitations of the synthetic experiment. On the other hand, being based on voting is what enables these algorithms to have an unsupervised mode of work (not requiring training datasets) and to be used for research purposes.

Actual accuracy measurement of the Truth Discovery algorithms wasn't possible due to the absence of ground truths data on the used three cell annotator claims. In the future, it will be useful to carry out a similar experiment with scRNAseq cell annotator claims accompanied by prior knowledge on ground truths. An example of how to acquire such dataset is to use scRNAseq cell typing in parallel with phenotypic cell identification (representing the ground truths).

It is important to note that prior data standardization of the cell annotators' claims has led to some loss of information due to generalization. As already mentioned in the Methods – more detailed cell type labels have been generalized to match simpler labels from other cell annotators (the SCN algorithm presented with the most general labels of the three).

5.3 Conclusions

Overall, the results showcased that the Truth Discovery algorithms are applicable to cell type data and show a promising potential for the proposed use case. The research project was successful in defining limitations and guidelines for further investigation which need to be considered for a successful, reliable implementation. To ensure minimal information loss during input data standardization, annotators with conserved labels and similar cell specificity must be selected for use with this methodology. Supervised cell annotators such as the three ones used in the real data experiment are more realistic to use as sources, because of their relatively standard labels as opposed to unsupervised cell annotators. Unsupervised cell annotators aren't as ubiquitous and there is even less possibility of labels standardization due to them dealing with higher level of complexity and uncertainty. The results also suggested that for Truth Discovery to be accurate enough, input datasets with as much sources as possible must be used. In the future it might be worth trying to assemble a dedicated experimental pipeline, where more than 10 supervised cell annotators with conserved cell type labels are used to annotate scRNAseq data, after which Truth Discovery is applied to the cell type predictions (maximizing potential accuracy of cell identification by accounting for each annotator).

6. References:

Abdelaal, T., Michielsen, L., Cats, D., Hoogduin, D., Mei, H., Reinders, M. J. T., & Mahfouz, A. (2019). A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biology*, *20*(1). https://doi.org/10.1186/s13059-019-1795-z

Aran, D., Looney, A. P., Liu, L., Wu, E., Fong, V., Hsu, A., Chak, S., Naikawadi, R. P., Wolters, P. J., Abate, A. R., Butte, A. J., & Bhattacharya, M. (2019). Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature Immunology*, *20*(2). https://doi.org/10.1038/s41590-018-0276-y

Berti-Équille, L., & Borge-Holthoefer, J. (2015). Veracity of Data: From Truth Discovery Computation Algorithms to Models of Misinformation Dynamics. *Synthesis Lectures on Data Management*, *7*(3). https://doi.org/10.2200/s00676ed1v01y201509dtm042

Bleiholder, J., & Naumann, F. (2009). Data Fusion. *ACM Computing Surveys*, *41*(1). https://doi.org/10.1145/1456650.1456651

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C., & Collins, J. J. (2018). Next-Generation Machine Learning for Biological Networks. In *Cell* (Vol. 173, Issue 7). https://doi.org/10.1016/j.cell.2018.05.015

Chu, Y., & Corey, D. R. (2012). RNA sequencing: Platform selection, experimental design, and data interpretation. *Nucleic Acid Therapeutics*, *22*(4). https://doi.org/10.1089/nat.2012.0367

Dong, X. L., Berti-Equille, L., & Srivastava, D. (2009). Integrating conflicting data: The role of source dependence. *Proceedings of the VLDB Endowment*, *2*(1). https://doi.org/10.14778/1687627.1687690

Hao, J., & Ho, T. K. (2019). Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language. In *Journal of Educational and Behavioral Statistics* (Vol. 44, Issue 3). https://doi.org/10.3102/1076998619832248

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W. M., Zheng, S., Butler, A., Lee, M. J., Wilk, A. J., Darby, C., Zager, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E. P., Jain, J., Srivastava, A., Stuart, T., Fleming, L. M., Yeung, B., … Satija, R. (2021). Integrated analysis of multimodal single-cell data. *Cell*, *184*(13). https://doi.org/10.1016/j.cell.2021.04.048

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. In *Nature* (Vol. 585, Issue 7825). https://doi.org/10.1038/s41586-020-2649-2

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., & Hemberg, M. (2017). SC3: Consensus clustering of single-cell RNA-seq data. *Nature Methods*, *14*(5). https://doi.org/10.1038/nmeth.4236

Kolodziejczyk, A. A., Kim, J. K., Svensson, V., Marioni, J. C., & Teichmann, S. A. (2015). The Technology and Biology of Single-Cell RNA Sequencing. In *Molecular Cell* (Vol. 58, Issue 4). https://doi.org/10.1016/j.molcel.2015.04.005

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A., & Robles, V. (2006). Machine learning in bioinformatics. In *Briefings in Bioinformatics* (Vol. 7, Issue 1). https://doi.org/10.1093/bib/bbk007

Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W., & Han, J. (2014). Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/2588555.2610509

Li, Z., & Feng, H. (2022). A neural network-based method for exhaustive cell label assignment using single cell RNA-seq data. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-021-04473-4

Ma, F., & Pellegrini, M. (2020). ACTINN: Automated identification of cell types in single cell RNA sequencing. *Bioinformatics*, *36*(2). https://doi.org/10.1093/bioinformatics/btz592

Ma, W., Su, K., & Wu, H. (2021). Evaluation of some aspects in supervised cell type identification for single-cell RNA-seq: classifier, feature selection, and reference construction. *Genome Biology*, *22*(1). https://doi.org/10.1186/s13059-021-02480-2

Mukherjee, S., Weikum, G., & Danescu-Niculescu-Mizil, C. (2014). People on drugs: Credibility of user statements in health communities. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2623330.2623714

Ozsolak, F., & Milos, P. M. (2011). RNA sequencing: Advances, challenges and opportunities. In *Nature Reviews Genetics* (Vol. 12, Issue 2). https://doi.org/10.1038/nrg2934

Pasternack, J., & Roth, D. (2010). Knowing what to believe (when you already know something). *Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference*, *2*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peyvandipour, A., Shafi, A., Saberian, N., & Draghici, S. (2020). Identification of cell types from single cell data using stable clustering. *Scientific Reports*, *10*(1). https://doi.org/10.1038/s41598-020-66848-3

Singleton, J. (2019). *truthdiscovery*. Cardiff University, School of Computer Science and Informatics.

Stegle, O., Teichmann, S. A., & Marioni, J. C. (2015). Computational and analytical challenges in single-cell transcriptomics. In *Nature Reviews Genetics* (Vol. 16, Issue 3). https://doi.org/10.1038/nrg3833

Stormo, G. D., Schneider, T. D., Gold, L., & Ehrenfeucht, A. (1982). Use of the “perceptron” algorithm to distinguish translational initiation sites in E. coli. *Nucleic Acids Research*, *10*(9). https://doi.org/10.1093/nar/10.9.2997

Tan, Y., & Cahan, P. (2019). SingleCellNet: A Computational Tool to Classify Single Cell RNA-Seq Data Across Platforms and Across Species. *Cell Systems*, *9*(2). https://doi.org/10.1016/j.cels.2019.06.004

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., Lao, K., & Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, *6*(5). https://doi.org/10.1038/nmeth.1315

Wang, D., Kaplan, L., Le, H., & Abdelzaher, T. (2012). On truth discovery in social sensing: A maximum likelihood estimation approach. *IPSN'12 - Proceedings of the 11th International Conference on Information Processing in Sensor Networks*. https://doi.org/10.1145/2185677.2185737

Wang, X., Sheng, Q. Z., Fang, X. S., Li, X., Xu, X., & Yao, L. (2015). Approximate truth discovery via problem scale reduction. *International Conference on Information and Knowledge Management, Proceedings*, *19-23-Oct-2015*. https://doi.org/10.1145/2806416.2806444

Wang, Z., Gerstein, M., & Snyder, M. (2009). RNA-Seq: A revolutionary tool for transcriptomics. In *Nature Reviews Genetics* (Vol. 10, Issue 1). https://doi.org/10.1038/nrg2484

Yin, X., Han, J., & Yu, P. S. (2008). Truth discovery with multiple conflicting information providers on the Web. *IEEE Transactions on Knowledge and Data Engineering*, *20*(6). https://doi.org/10.1109/TKDE.2007.190745

Zhao, B., Rubinstein, B. I. P., Gemmell, J., & Han, J. (2012). A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment*, *5*(6). https://doi.org/10.14778/2168651.2168656

Zhi, S., Zhao, B., Tong, W., Gao, J., Yu, D., Ji, H., & Han, J. (2015). Modeling truth existence in truth discovery. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *2015-August*. https://doi.org/10.1145/2783258.2783339