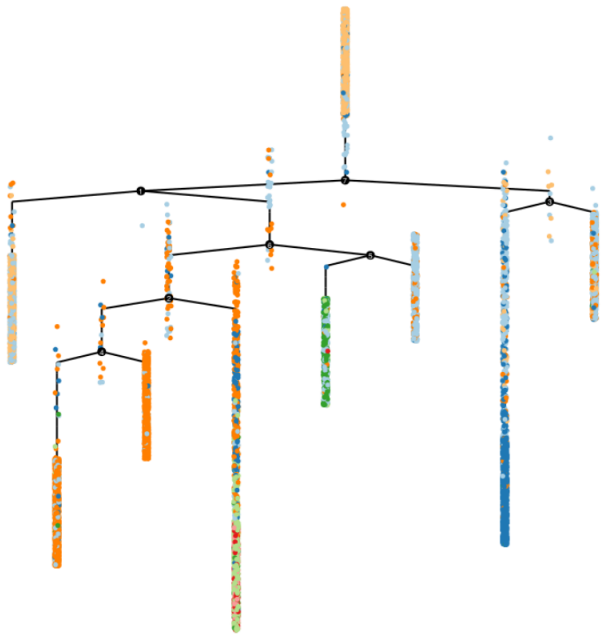
CLUSTER MACHINE LEARNING TO REVEAL CELL IDENTITIES



SUNDAY 27TH OCTOBER 2019

PATRICK KNOTT

Supervisors:

Dr. Mohammad Ali Moni (USYD)

Dr. Fida Hasan (QUT)

1. Executive Summary

Single-cell RNA-sequencing is a new technique for medical and biological research. It has led to significant advances in knowledge, improved methods for medical diagnosis and superior medical treatments with more effective and targeted results.

A fundamental step in single-cell RNA-sequencing involves the identification of similarities between cells using mathematical techniques. As the field is still in its infancy, no best-practices have been determined for either the maths to use or their optimal software implementation.

This project sought to develop and analyse some new methods for the classification of cells into clusters of resemblance. It sought to create computational templates to supplement existing pipelines for single-cell RNA-sequencing analysis. It implemented new techniques for gene-selection, dimensionality reduction and clustering algorithms.

Medical researchers and molecular biologists can insert these templates into their existing workflow to deepen their understanding of the structure and behaviour of cells within a tissue sample, leading to better diagnostic techniques, more targeted treatments for illnesses, and fewer side-effects from those treatments.

It found strong evidence that some of these new tools reveal information about tissue samples which remains hidden to existing tools.

TABLE OF CONTENTS

List Of Tables

List Of Figures

1.0 INTRODUCTION

Single-cell RNA-sequencing (scRNAseq) is a revolutionary technique that allows investigation of the fundamental biological unit, the cell (Haque, 2017). Shifting the focus of research from the aggregate effects of cell samples in their entirety to the multi-tiered complexity of individual cells (Menden, 2019), can reveal the underlying structure and behaviour of biological processes (Ntranos, 2016). This technology is still in its infancy so there are no agreed upon best-practices or standardized software for scientists to use (Luecken, 2019).

Developing new tools and methods to discover these cell-identities is the goal of this project.

This project found that different choices for pre-processing technique, gene-selection technique, dimension reduction technique, clustering algorithm and the number of clusters can have dramatic effects on the resulting cluster labels. No particular combination of parameters demonstrated superiority across the validation techniques. Instead, it appears that combinations of parameters amplified and hid different aspects of the underlying structure. This report found strong evidence to suggest that the field would benefit from using a wider range of mathematical analysis techniques.

1.1 ENVIRONMENTAL SCAN

scRNAseq is only ten years old, so the best-practice processes are still an open question. There have been many significant advances in the laboratory steps of the process recently, resulting in larger data sets with more accurate details

(Lafzi, 2018). The software needs of the medical and scientific experts are evolving too (Kiselev, 2019).

Ribonucleic acid, or RNA, is essentially a blueprint used to create a functional product, usually a protein. Each cell contains tens of thousands of activated genes, which are pieces of DNA, and each gene creates many RNA molecules. scRNAseq quantifies the specific RNA within each cell. This is in contrast to the older technique of bulk sequencing, in which the components of each of the many cells in a tissue sample are mixed together. There are no clear limits between the types and behaviours of cells, so clustering techniques are used to identify similar cells, called cell-identities (Luecken, 2019).

Two of the most common software pipelines in use for scRNAseq are Luecken and Seurat (Luecken, 2019; Stoeckius, 2018). The details of most of the many differences are outside the scope of this project.

What is relevant is that they both use the same gene selection tools (Cell-ranger and Seurat dispersions), both use principal component analysis for dimensionality reduction, and both use Louvain clustering (Blondel, 2008) to reveal cell identities.

This project sought to develop modular software to insert within the Luecken and Seurat pipelines, using alternatives for these three tasks: gene selection, dimensionality reduction and clustering.

1.2 MOTIVATION

ScRNAseq has already led to big advances in medical and biological understanding. Some of its biggest success stories have been with cancer. Comparing the different gene expression profiles of cell identities between healthy and cancerous patients can uncover biomarkers for cancer. For example, haematopoietic scRNAseq was used to develop a 17-gene biomarker test to separate acute myeloid leukaemia patients into cohorts requiring differing treatment plans (Zheng, 2018).

As well as aiding in diagnosis, scRNAseq also gives singular detail about the behaviour of cancer. There are many functions a cancer growth must perform for it to be successful, e.g. attract blood vessels with angiogenesis and suppress the secretion of adhesive molecules (Ding, 2018). scRNAseq allows development of tools to target particular cancer functions, resulting in a more direct effect and fewer side-effects (Chung, 2017).

Another recent example was a step towards the possibility of a cancer vaccine (Shen, 2019). A team from Arizona State University, using scRNAseq, discovered 200,000 new neo-antigens that are common to multiple tumour types. Only about 40% of tumours contain enough mutations in their DNA to make a vaccine, but the team discovered enough mutations in the RNA of every sample they sequenced.

Every year in Australia alone, cancer results in 50 000 deaths, 145 000 diagnoses, and $4.5 billion in direct medical costs (Cancer Australia, 2019). Even a small improvement in knowledge or treatment for cancer would have an enormous benefit to public health and quality of life, and also to the economy.

1.3 OBJECTIVE

The primary objective of this project is to develop software tools for use in scRNAseq workflows. The code was developed to insert after either of the two most common pre-processing workflows, Luecken or Seurat

A secondary objective was to investigate the results and infer if this software is of any benefit. The validity of a clustering attempt can only be determined by successful gene-annotation (Luecken, 2019), and this researcher has no specialist medical or biological training. But biologically significant clustering usually has good statistical properties (Brock, 2008). So, while these results are only evaluated from a mathematical perspective, that can be a useful and informative first step.

1.4 SINGLE-CELL RNA SEQUENCING PROCESS

The single cell RNA sequencing process can be thought of as a pipeline. It starts in a laboratory then shifts to a computer. Most steps in the procedure are performed by a domain specialist, but some of them require mathematical and computational expertise.

scRNAseq:

- take tissue sample
- isolate the individual cells
- burst each cell to reveal its contents
- apply unique barcode to all RNA from a single cell
- merge the contents of all the burst cells
- run them through a sequencing machine

- output into matrices of individual cells versus expressed gene counts
- quality control e.g. remove the after effect of cells which burst before the cell isolation, contaminating other cells
- data normalization e.g. convert absolute gene counts to counts-per-million or log-normalization
- data correction e.g. account for batch effects

The processing above requires expertise in the relevant scientific field to make informed decisions (Lafzi, 2018), so these steps have been duplicated from the tutorials for Luecken and Seurat pipelines. The next few steps can be performed by the domain specialist, or the mathematical specialist (data scientist, computer scientist, biostatistician, bio-informaticist), or both:

- select which genes to include in the analysis
- dimensionality reduction
- clustering

It is these three steps (gene selection, dimensionality reduction and clustering) which this project is concerned with.

After the clusters have been determined, the analyses are performed. These use the full data set, not the dimensionally reduced data, but retain the clustering labels generated with the reduced data. The three analyses are:

- differential expression (to compare which genes are expressed in different samples, for example between a healthy patient and an unhealthy patient, or the one patient but at different points in time)
- trajectory inference and gene dynamics (which attempts to factor in time, and the lifecycles of individual cells)

- compositional analysis (which relates to the proportions of total cells that fall into each cell-identity cluster)

2.0 KEY DELIVERABLES

The deliverables for this project are a set of five Jupyter notebook templates. Researchers need to make only a few alterations to add them to their existing workflow:

1. perform and display results of new gene selection alternative, save selected genes
2. perform autoencoding to reduce dimensions, then cluster the reduced data set, display and save details of results
3. perform singular value decomposition for dimension reduction, then cluster the reduced data set, display and save details of results
4. perform sub-clustering of a chosen model, display and save details of results
5. combine results of sub-clustering, perform comparisons with the default Louvain cluster outputs

Table 1. Deliverables overview. The bolded techniques are those believed to be new applications to scRNAseq. Note that for each combination of the first three steps, each component of the last four steps was implemented

STEP	STEP NAME	TECHNIQUE
1	Pre-processing	Luecken or Seurat

STEP	STEP NAME	TECHNIQUE
2	Gene selection	Cell-ranger or Seurat dispersion calculation with automatic or **manual selection of highly variable gene threshold**, or all genes
3	Dimension reduction	**Singular value decomposition** or **auto-encoder**
4	Clustering	**K-means**, **hierarchical**, **Model-based**, **Self-Organizing Tree Algorithm**, **Partitioning Around Medoids**, **Clara**, **Diana**, **Fuzzy clustering**, **Gaussian Mixture Models**
5	Internal validation	**Connectivity**, **Dunn Index**, **Silhouette Width**
6	Stability validation	**Average Distance**, **Average Proportion of Non-Overlap**, **Average Distance Between Means**, **Figure of Merit**
7	Comparisons	**Akaike Information Criterion**, **Bayesian Information Criterion**, **average cluster power**, **Adjusted Rand Index**, **Normalized Mutual Information**, **Variation of Information**

2.1 System Overview

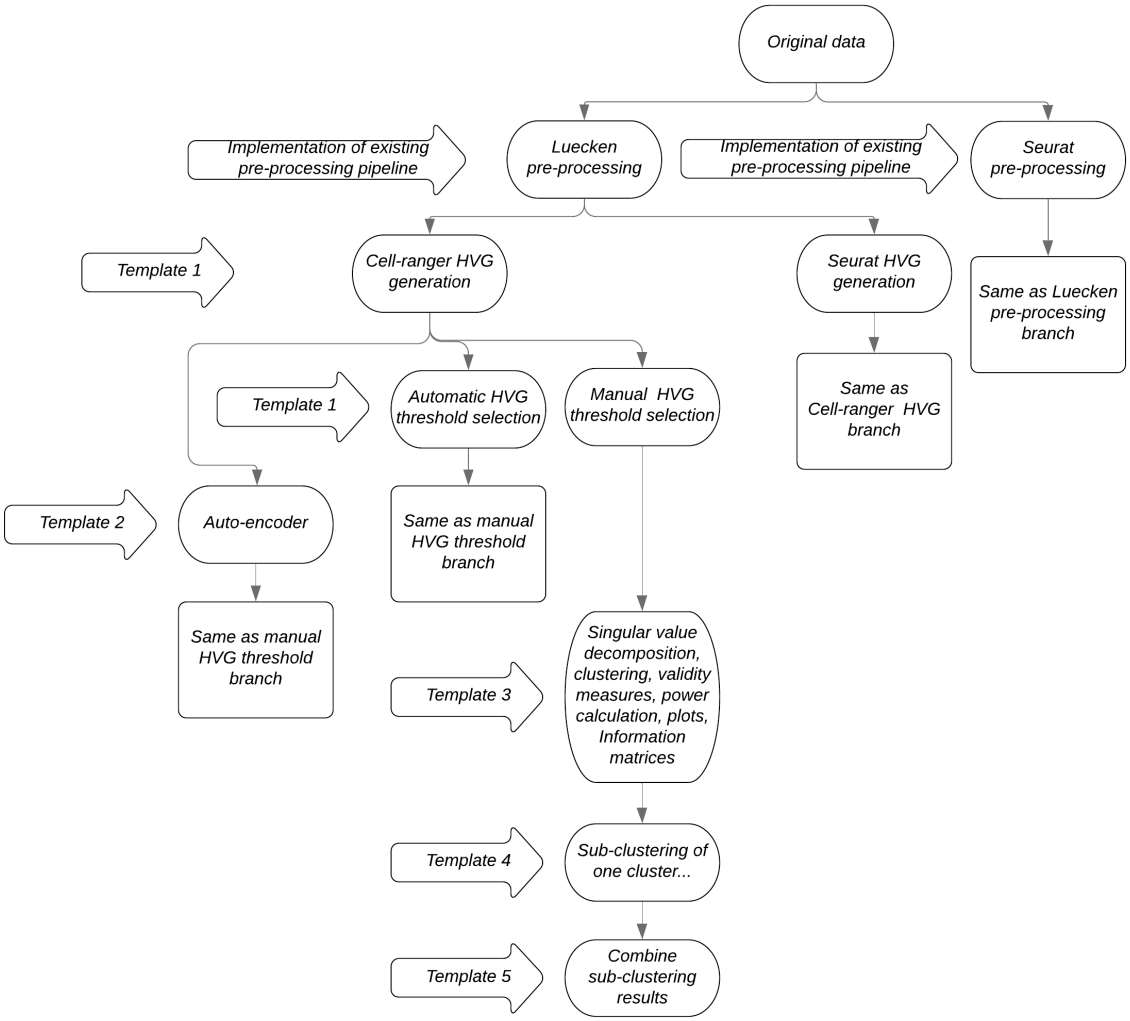


Figure 1. Dataflow overview

Table 2. Template overview

Template	User Inputs	Functionality	Outputs
0	• folder containing source data	• perform Luecken or Seurat pre-processing	• compressed ScanPy adata object • Louvain cluster labels

Template	User Inputs	Functionality	Outputs
1	• folder for inputs and outputs • adata file	• perform Seurat and Cell-ranger dispersion calculations on the data • save automatically selected highly variable genes (HVG) • plot dispersions then make manual selections of HVG thresholds based on 'elbows' • save list of all genes for comparison • perform Louvain clustering and save the list of cluster labels	• Boolean lists to identify different HVG selections
2	• folder for inputs and outputs • adata file	• trains autoencoder on the full data set for either Luecken or Seurat pre-processing	• the embedding of each cell into the auto-encoder bottleneck vector space
3	• folder for inputs and outputs • adata file • one Boolean list containing a selection of HVG • number of cells to	• perform singular value decomposition • plots eigenvalues for user to select 'elbow' as the number of eigenvectors to keep • calls R package clValid	• reduced matrix of eigenvectors • save outcome of validation metrics • save lists of cluster memberships for

Template	User Inputs	Functionality	Outputs
3 cont.	keep in model (for computational expense) • minimum and maximum number of clusters to build for each cluster algorithm	to perform all clustering algorithms except Gaussian Mixture Models (GMM), on range of number of clusters • calls clValid to perform all internal and stability validity measures • build GMM of same range of cluster numbers • calculate and plot the average power for each model • calculate and plot AIC and BIC for k-means and GMM models • compare models with Adjusted Rand Index, Mutual Information Index and Variation of Information with plots and data frames of all pairwise comparison	each cell for each model • save plots and data frame of average power for each model • save plots and data frames of AIC and BIC • save plots and data frames of Information theory comparisons
4	• folder for inputs and outputs • one Boolean list containing a selection of HVG	• repeats the whole clustering process from Template 3 on one cluster from the input cluster labels i.e. nine cluster	• save outcome of validation metrics • save lists of cluster memberships for

Template	User Inputs	Functionality	Outputs
4 cont.	• minimum and maximum number of clusters to build for each cluster algorithm • list of lists of cluster labels for all models from the corresponding Template 3 • the algorithm name and number of clusters to use for this sub-clustering process	algorithms, range of cluster numbers, internal validation, stability validation, average power, AIC, BIC, and three information theory comparisons	each cell for each model • save plots and data frame of average power for each model • save plots and data frames of AIC and BIC • save plots and data frames of Information theory comparisons
5	• folder for inputs and outputs • one Boolean list containing a selection of HVG • list of lists of cluster labels for all models from the corresponding Template 3 • the algorithm name and number of clusters to use	• combines the sub-clustering partition results into one • calculates information theory comparisons between Louvain clusters, one clustered data and twice clustered data	• none

Template	User Inputs	Functionality	Outputs
5 cont.	for this sub-clustering process		

2.2 CLUSTER EVALUATION

ScRNAseq cluster validation consists of three types of measures: internal, stability and biological (Brock, 2008). The first two are entirely mathematical, the third requires a domain specialist so it will not be performed. No measure is of primary importance. They investigate different characteristics of the clustering.

Internal validation measures (see Figure 2) relate to the similarities within and differences between clusters:

1. Dunn: maximize (0,∞), ratio of smallest inter-cluster distance between any two observations, to the largest intra-cluster distance between any two observations
2. Connectivity: minimize [0,∞), based on how many of k-neighbours are in different clusters
3. Silhouette Width: maximize (-1,1), ratio of the difference between an observation's similarity to its own cluster and its dissimilarity to all observations in a different cluster

```
Clustering Methods:
 kmeans hierarchical model sota pam clara diana fanny

Cluster sizes:
 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Validation Measures:
                                  3        4        5        6        7        8

kmeans       APN              0.2404   0.3689   0.3455   0.2737   0.2134   0.2638
             AD               0.1236   0.1220   0.1194   0.1156   0.1108   0.1097
             ADM              0.0178   0.0249   0.0256   0.0245   0.0173   0.0225
             FOM              0.0212   0.0211   0.0210   0.0209   0.0205   0.0202
             Connectivity   901.1929 870.9698 814.8599 741.7952 731.2881 955.2877
             Dunn             0.0997   0.0871   0.1070   0.1095   0.1095   0.0777
             Silhouette       0.1117   0.1168   0.0846   0.0951   0.1228   0.1257
hierarchical APN              0.0025   0.0043   0.0072   0.0105   0.0178   0.0229
             AD               0.1254   0.1250   0.1249   0.1247   0.1246   0.1243
             ADM              0.0018   0.0015   0.0018   0.0019   0.0026   0.0028
             FOM              0.0213   0.0213   0.0213   0.0212   0.0212   0.0210
             Connectivity    12.2226  17.6806  20.6095  23.7385  26.6675  35.2044
             Dunn             0.2421   0.2554   0.2554   0.2554   0.2554   0.2554
             Silhouette       0.3117   0.2884   0.2675   0.2359   0.2046   0.1856
model        APN              0.0465   0.1065   0.2531   0.3007   0.2164   0.2516
             AD               0.1192   0.1179   0.1173   0.1153   0.1129   0.1116
             ADM              0.0037   0.0074   0.0164   0.0185   0.0171   0.0171
             FOM              0.0203   0.0201   0.0200   0.0197   0.0195   0.0189
             Connectivity   417.5698 585.3341 672.0889 734.7988 999.0718 1013.7702
             Dunn             0.0835   0.0834   0.0834   0.0818   0.0834   0.0703
             Silhouette       0.0636   0.0496   0.0667   0.0739   0.0907   0.0617
sota         APN              0.3766   0.3888   0.3716   0.3791   0.3908   0.3228
```

Figure 2. Partial example of internal and stability validation output

Stability validation measures (see Figure 2) relate to how the clustering changes if one feature is removed. The scores are the averaged value for each feature removal:

1. Average proportion of non-overlap: minimize [0,1), proportion of observations not in same cluster in both models
2. Average distance: minimize (0,∞), average distance between observations in the same cluster in both models
3. Average distance between means: minimize (0,∞), average distance cluster means move when one feature is removed
4. Figure of Merit: minimize [0,∞), average intra-cluster variance of deleted feature

The following measures can be used to decide which model is better

1. Average power: maximize (0,1), average of power for each feature in each cluster for one cluster method and number of clusters (see Figure 5).
2. Akaike Information Criterion (altered): minimize (0,∞), residual sum of squares (instead of negative log likelihood) plus the number of clusters (Fee figure 7).
3. Bayesian Information Criterion (altered): minimize (0,∞), residual sum of squares (instead of negative log likelihood) plus the number of cluster times the number of features (Fee figure 8).

The following measures compare two model's outputs to each other, but don't evaluate the quality of a model

1. Adjusted Rand Index: [0,1], the number of pairs of observations which are either in the same cluster as each other in both models or in different clusters as each other in both model, divided by the number of pairs of observations in the whole set, then adjusted to incorporate chance (see Figure 24)
2. Normalized Mutual Information: [0,1], proportion of total information shared between the two models (see Figure 3)
3. Variation of Information: [0,∞), amount of total information not shared between two models, not scaled (see Figure 9)

	kmeans2	kmeans3	kmeans4	kmeans5	kmeans6	kmeans7	kmeans8
kmeans2	0	0.55548054	0.47360677	0.423168	0.35460733	0.31930786	0.34872551
kmeans3	0.55548054	0	0.8733932	0.68170021	0.52342963	0.49973102	0.48697518
kmeans4	0.47360677	0.8733932	0	0.65628533	0.5174931	0.52983436	0.50193542
kmeans5	0.423168	0.68170021	0.65628533	0	0.73631544	0.66666254	0.64674587
kmeans6	0.35460733	0.52342963	0.5174931	0.73631544	0	0.57185818	0.61676021
kmeans7	0.31930786	0.49973102	0.52983436	0.66666254	0.57185818	0	0.90339978
kmeans8	0.34872551	0.48697518	0.50193542	0.64674587	0.61676021	0.90339978	0
kmeans9	0.32677488	0.44110027	0.45796394	0.61056781	0.68054505	0.78961009	0.85864223
kmeans10	0.42067424	0.54850445	0.54127433	0.64744327	0.64519429	0.71601158	0.76820177
kmeans11	0.40794759	0.53629219	0.52944504	0.65679731	0.65363531	0.72288153	0.77403887
kmeans12	0.34956767	0.52401536	0.51272751	0.651736	0.67282101	0.70342999	0.75868388
kmeans13	0.34028825	0.52615324	0.51776305	0.63373315	0.61799019	0.68182651	0.73789709
kmeans14	0.3931718	0.52645171	0.52172476	0.66831007	0.67894997	0.61237792	0.66861702
kmeans15	0.38696225	0.51742869	0.51328358	0.65638194	0.64827283	0.65322704	0.70046849

Figure 3. Part of a matrix of pair-wise Normalized Mutual Information (the diagonal zero values are an artefact of the table generation, not a value)

2.3 LIMITATIONS

1. computational resources (Figure 4)
2. lack of domain specialist knowledge

Call cluster algorithms in R

In [7]:

```
%%R -i xr -i data_save_loc -i data_filename -i k_min -i k_max -i k_step -o cl_obj_eucl


clmethods <- c("kmeans", "hierarchical", "model", "sota", "pam", "clara", "diana", "fanny")
vlmethods = c("internal", "stability")
cl_obj <- clValid(xr,
                        nClust = seq(k_min,k_max,k_step),
                        clMethods = clmethods,
                        validation = vlmethods,
                        maxitems = 20000,
                        metric = "euclidean",
                        verbose = TRUE)


cl_obj_eucl = cl_obj
clv_file_name =paste(data_save_loc, "cl_obj_eucl_", data_filename, ".rds")
plot_file_name =paste(data_save_loc, "cl_obj_eucl_", data_filename, ".jpg")
summary_file_name =paste(data_save_loc, "cl_obj_eucl_SUMMARY_", data_filename, ".txt")
sink(summary_file_name)
print(summary(cl_obj))
sink()
saveRDS(cl_obj, file=clv_file_name)


jpeg(plot_file_name)
plot(cl_obj)
dev.off()

```

executed in 19h 28m 31s, finished 07:31:41 2019-10-15

Figure 4. Jupyter notebook cell. Red rectangle in lower left edge showing that the runtime for twenty-two cluster sizes, eight cluster algorithms and both the internal and stability validation, was more than nineteen hours.

2.4 SUBSTANTIAL CHANGES FROM INITIAL SCOPE:

1. The original intent was to use new clustering techniques to uncover signature genes for cancer. The computational expense of performing the previously stated clustering options on multiple different data sets was too much for the resources available.
2. Manhattan distance and correlation were not investigated as alternative distance metrics to Euclidean
3. Automated cluster annotation was removed from the product backlog

3.0 PROJECT MANAGEMENT

This project was managed as an Agile scrum process. This is common for software development as it is inherently difficult to estimate development time or to predict roadblocks. This project was also limited by a strict timeframe, had no capacity to add more researchers, and only had domestic computational resources available. As such, the only variable in the project management was the product backlog.

The project had two supervisors, Dr. Hasan from Queensland University of Technology and Dr. Moni from The University of Sydney. There were four other QUT students working on the same topic (each individually). So, meetings were held as a group on campus with Dr. Moni on teleconference. Intermittent communication used email so that both supervisors could be involved.

The computational restrictions ended up being significant. A 2014 MacBook Pro with 2.5 GHz Intel Core i7 and 16GB RAM was used. The software used Jupyter Notebooks running Python 3.7, with most of the clustering and validation performed in calls to R 3.5.0. Python package ScanPy performed most of the pre-processing, with R package clValid the clustering and validation.

The data set used for the example templates was of epithelium cells from a mouse intestine. After each of the two pre-processing pipelines, there were about 13 000 cells remaining. This was beyond the computational resources available. So, only 2000 cells were used.

To further decrease the computational expense, the only distance metric used for clustering was Euclidean distance. Manhattan distance and correlation were removed from the product backlog. Ensemble modelling techniques were used only for stability validation and not for cluster optimisation. Ensemble clustering has shown great promise (Ronan, 2017), but has a multiplicative effect on processing time (Stuart, 2019).

Despite these reductions in computation, the set of eight clValid clustering algorithms, three internal validation algorithms, and the four stability validation took nearly twenty hours to run (see Figure 4). The RAM requirements meant that the computer could only run two notebooks at once, and even that required that the computer was not used for anything else.

3.1 PROJECT METHODOLOGY

The methodology to output each of the deliverables of this project consisted of:

1. Research Current Practice

Investigating the existing workflows for scRNAseq and of the mathematical tools more generally. The machine learning concepts used in this project are not new tools but new applications of existing tools.

2. Implementation

Turning the ideas into software, and adapting them to connect after the Luecken and Seurat tutorial pre-processing workflows

3. Evaluation (and Recommendation)

This researcher has no specialist medical knowledge, but there are many mathematical metrics available to gauge the validity of a clustering output.

Notable instances and trends in the validity measures will be given a superficial analysis in 666.666 Results and Discussion.

4.0 Results & Discussion

The purpose of this project was to create software. Complete evaluation of scRNAseq cluster requires domain specialist knowledge (Luecken, 2019), which was not available. However, statistical properties are often a strong indicator of biological clustering quality and suitability (Brock, 2008). So, the outcomes of the implementation of this software will be investigated as reasonable proxy for the usefulness of this software to medical and biological specialists.

Choices for pre-processing technique, dispersion calculation method, gene-selection number, cluster algorithm and number of clusters often had a significant effect on the validation metrics. But sometimes there was no discernible difference.

Figure 5 and 6 show the average power of each cluster algorithm at each prescribed number of clusters (Note that the vertical scales are not the same). The only difference between the two sets of models is the input data. Both use an auto-encoded fifteen-dimension reduction of the data set, but Figure 5 used the Luecken pre-processing and Figure 6 used the Seurat pre-processing. With the former, for most of the models the power is in the range of 35-55%. But the latter are 5-20%. This drastic difference is solely due to the choice of pre-processing workflow.

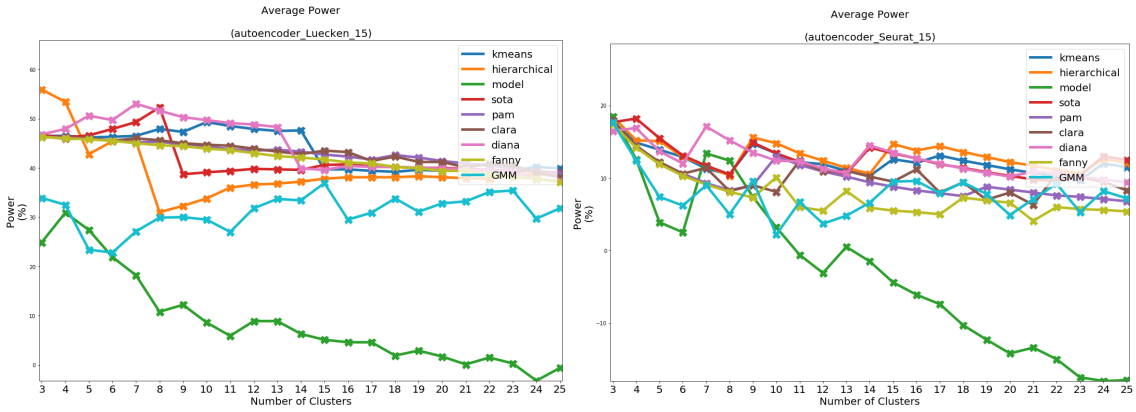


Figure 5 and 6. Average power, auto-encoded, 15 dimension, Luecken and Seurat pre-processing respectively.

Figure 7 shows the Akaike Information Criterion for k-means and Gaussian Mixture Models and Figure 8 shows the Bayesian Information Criterion for k-means and GMM (Note that only k-means and GMM could be evaluated with AIC and BIC). The absolute values don't matter, it's the relative values which compare the quality of different models. For both pre-processing methods, and all gene selections, it was usually the case that k-means was better than GMM. It also tended to be in the seven to ten cluster range the maximum number of clusters built. To a layman, this was surprising. K-means creates spherical clusters, whereas GMM creates clusters where the shape is a combination of Normal distributions. So many biological phenomena are Normally distributed, it seems logical that the distribution of cellular characteristics across many cells would be of similar distributions.

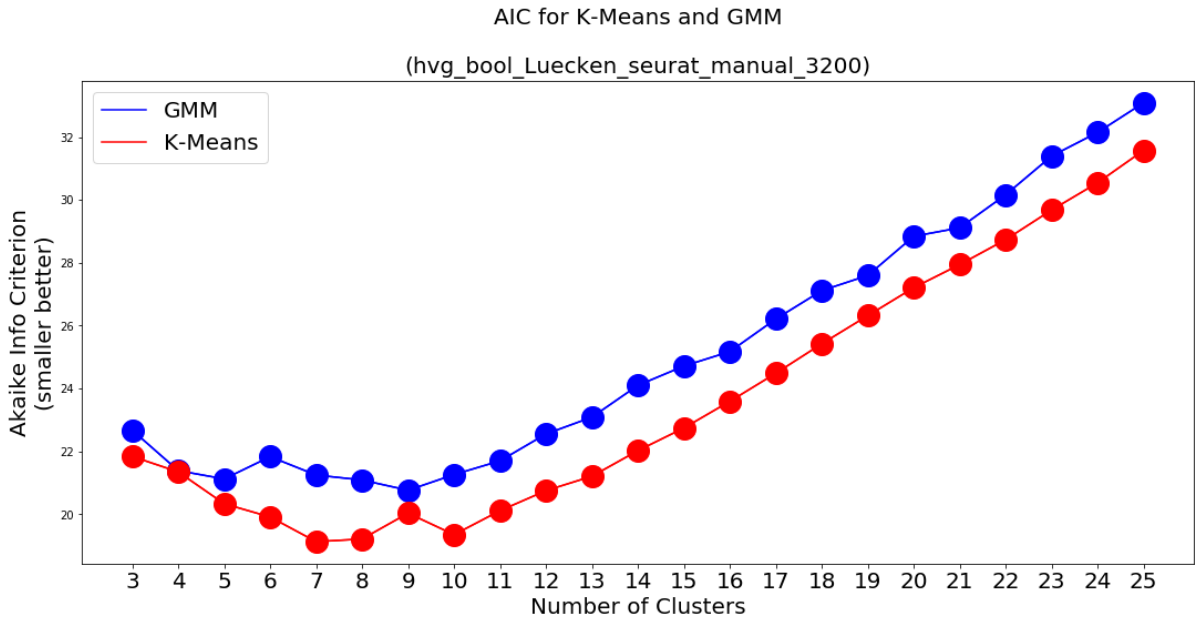


Figure 7. AIC Luecken pre-processing, Seurat dispersions, 3200 genes

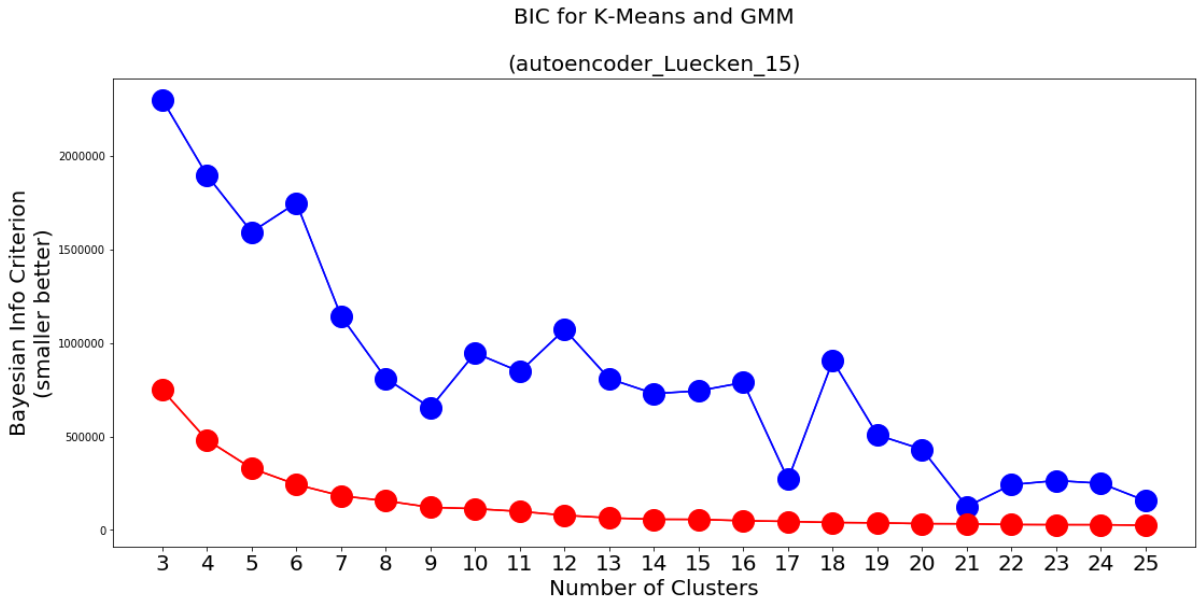


Figure 8. BIC Auto-encoded, Luecken pre-processing, 15 dimension

Figure 9 shows the average Variation of Information of each cluster method from one model, and Figure 10 the Variation of Information when one of its clusters are sub-clustered. In both plots the cluster methods have very different results. Apart from hierarchical, SOTA and Diana in the first plot, the number of clusters has a big impact too.

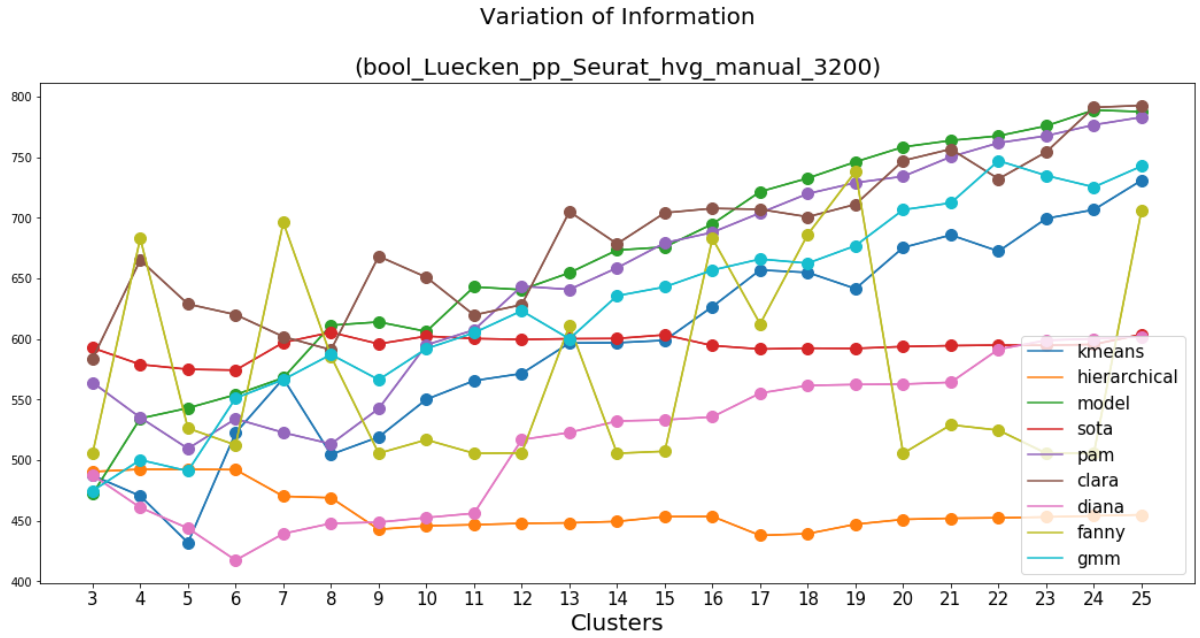


Figure 9. Variation of Information Luecken pre-processing, Seurat dispersions, 3200 genes

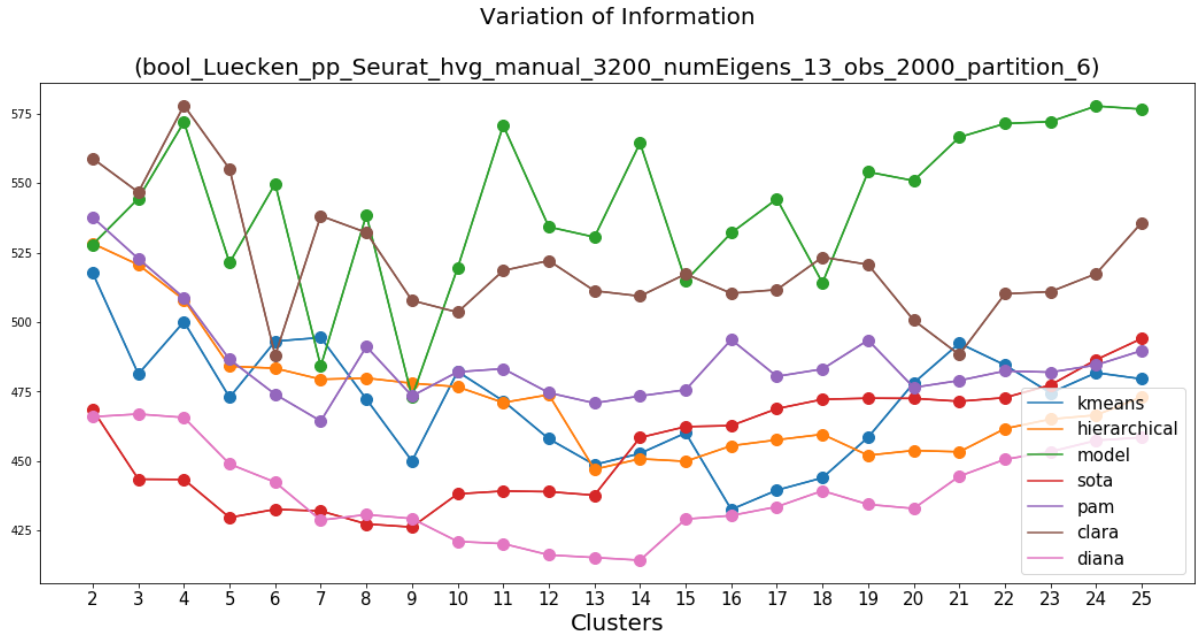


Figure 10. Variation of Information, Luecken pre-processing, Seurat dispersions, 3200 genes, partition 6

In Figure 11 the number of clusters has a significant impact on the Figure of Merit.

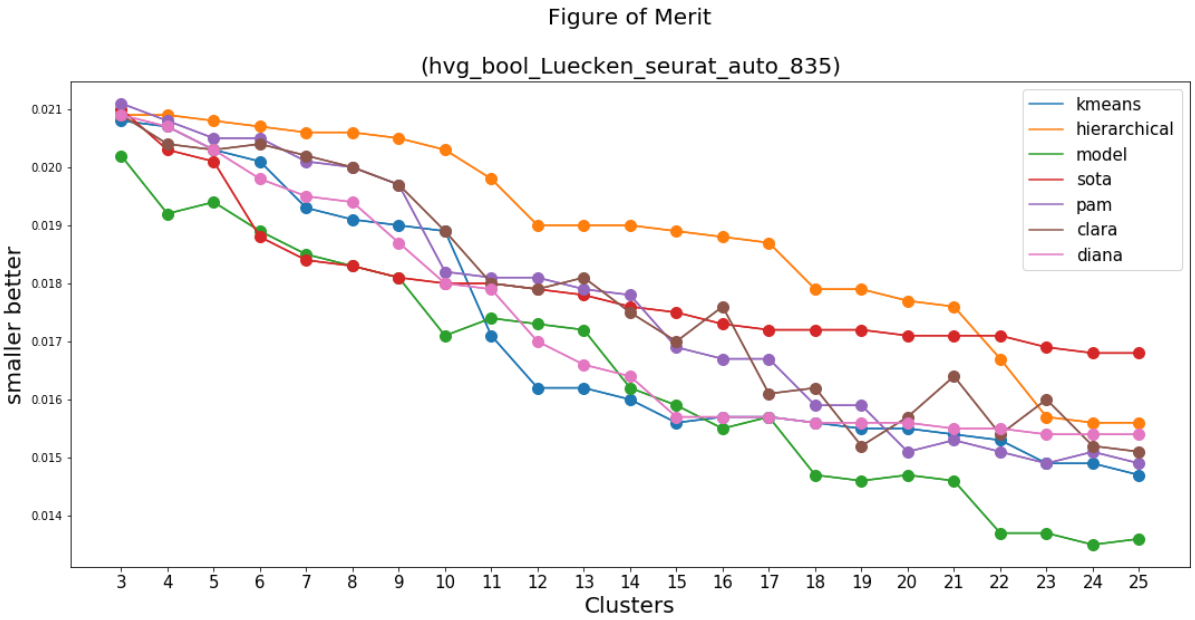


Figure 11. Figure of Merit, Luecken pre-processing, Seurat dispersions, 3200 genes

However, in Figures 12 and 13 the number of clusters had an inconsistent effect. The plots show the Average Distance Between Means for the full set of Luecken genes and the fifteen dimension, auto-encoded Seurat genes. This is perhaps supportive of using more than one layer of clustering. If the number of clusters has little impact that could be because the inherent structure of the biological structure is that of multiple layers of partitioning.

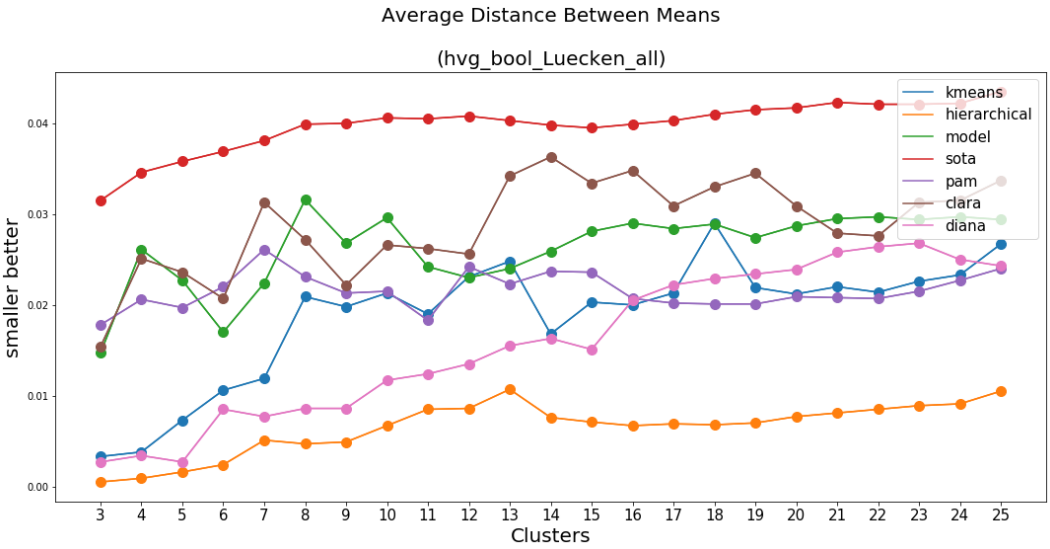


Figure 12. Average Distance Between Means, Luecken pre-processing, all genes

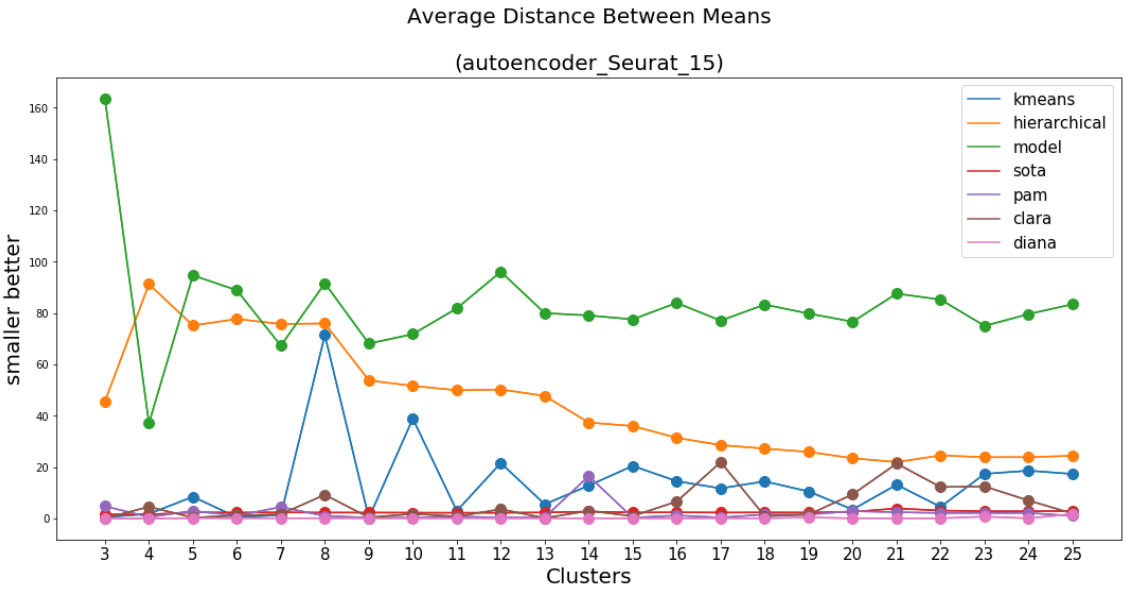
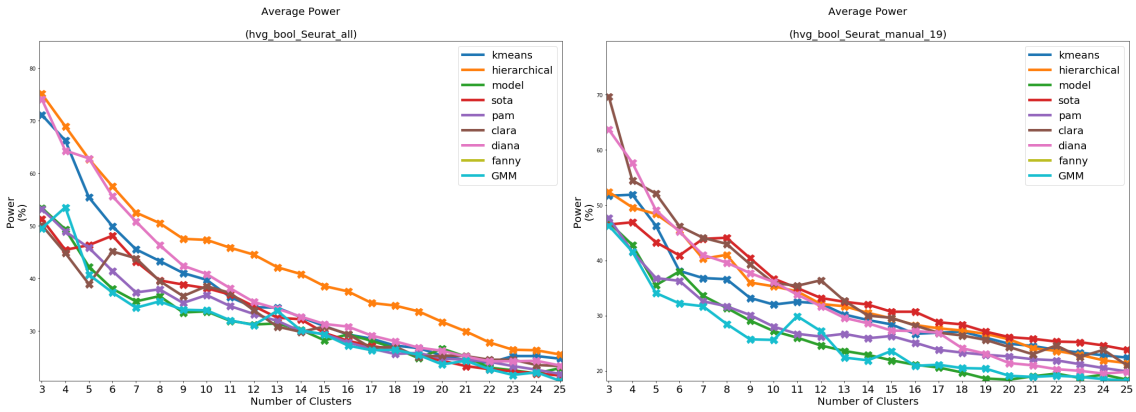


Figure 13. Average Distance Between Means, Seurat pre-processing, auto-encoded, 15 dimension

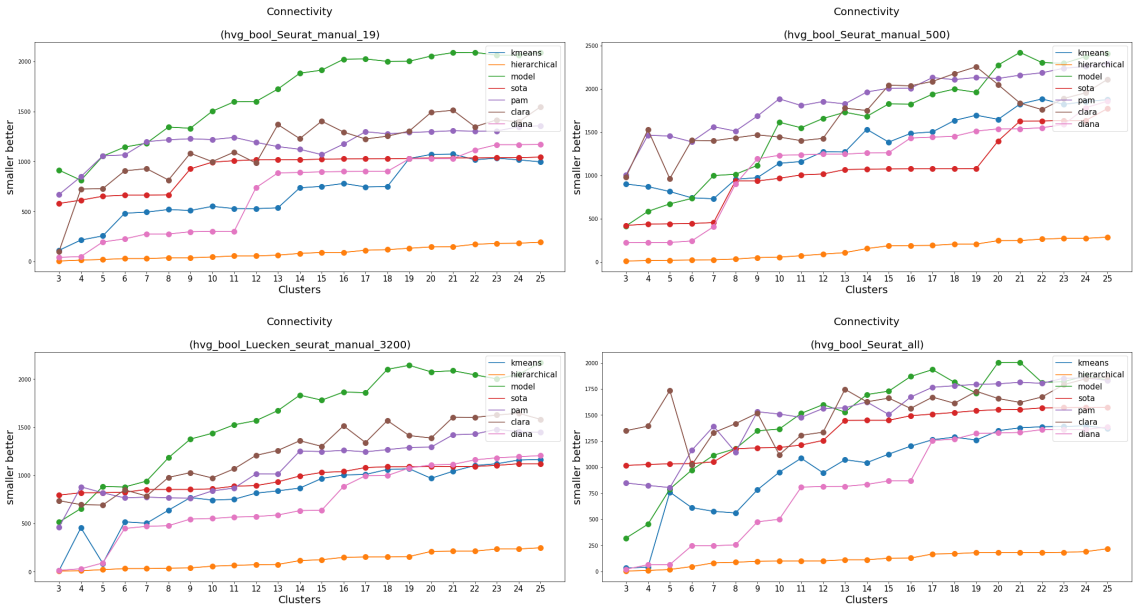
Figures 14 and 15 both use Seurat pre-processing. The left figure inputs all 13317 genes into the dimensionality reduction, but the right inputs only 19. Yet, the power of the sets of models are essentially the same. This is surprising because so much prior research has been focussed on gene-selection. It is

possible this behaviour is specific to this dataset, or it may imply that it is sometimes the case with other data too.



Figures 14 and 15. Average power, Seurat pre-processing, all genes and 19 genes respectively

Figures 16, 17, 18 and 19 show the Connectivity scores for four models. The only difference between the models is the number of genes, ranging from 19 to 13317, yet they have very similar scores for each algorithm and number of clusters



Figures 16, 17, 18 and 19. Connectivity, Luecken pre-processing, Seurat dispersions, with 19, 500, 3200 and all genes respectively

Figure 20 shows that the number of clusters had little impact on the Dunn index for one pre-processing and gene-selection input.

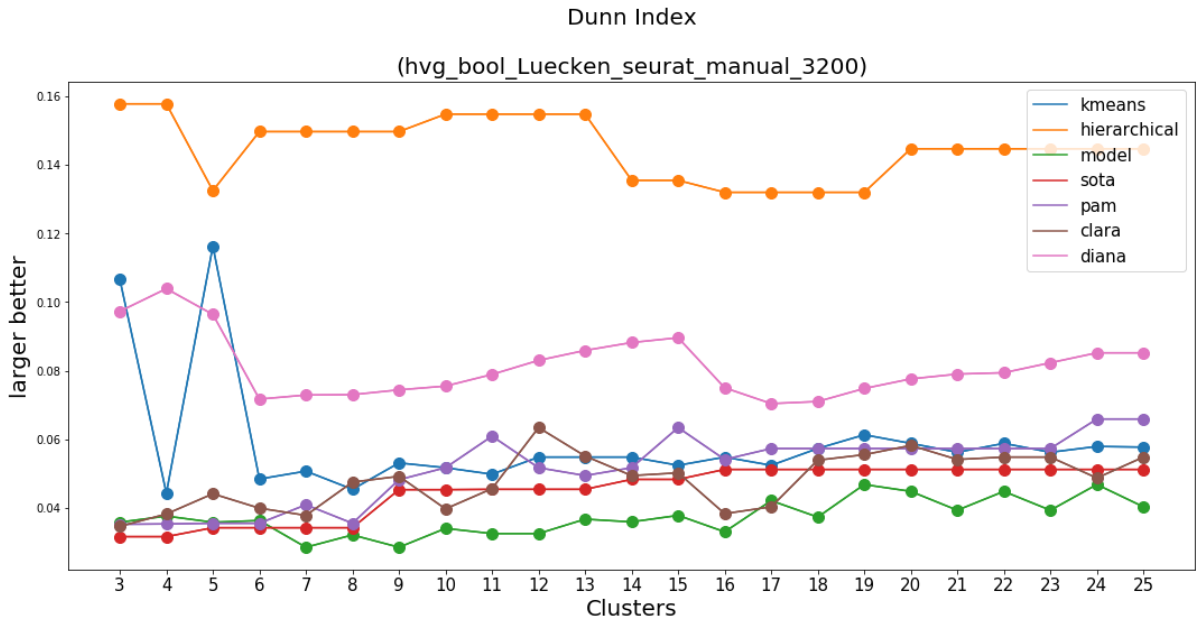


Figure 20. Dunn Index, Luecken pre-processing, Seurat dispersions, 3200 genes

For most pre-processing and gene-selection inputs, (agglomerative) Hierarchical scored the best on the internal and stability validation measures (Figures 21, 22). Interestingly, the divisive hierarchical method (Diana) usually did not rate near the best models.

Optimal Scores:

	Score	Method	Clusters
APN	0.0024	hierarchical	3
AD	0.0655	pam	25
ADM	0.0010	hierarchical	3
FOM	0.0141	model	25
Connectivity	6.0881	hierarchical	3
Dunn	0.1811	hierarchical	4
Silhouette	0.5304	hierarchical	3

Optimal Scores:

	Score	Method	Clusters
APN	0.0025	hierarchical	3
AD	0.0978	kmeans	25
ADM	0.0015	hierarchical	4
FOM	0.0169	model	25
Connectivity	12.2226	hierarchical	3
Dunn	0.2554	hierarchical	4
Silhouette	0.3117	hierarchical	3

Figure 21 and 22. Optimal models for validation methods for the Luecken pre-processing using Seurat dispersion to pick 3200 genes, and Seurat pre-processing using Cell-ranger dispersion to pick 500 genes.

The Model-based cluster method was often the lowest scored and least similar to other models, as seen in Figures 23 and 24 respectively.

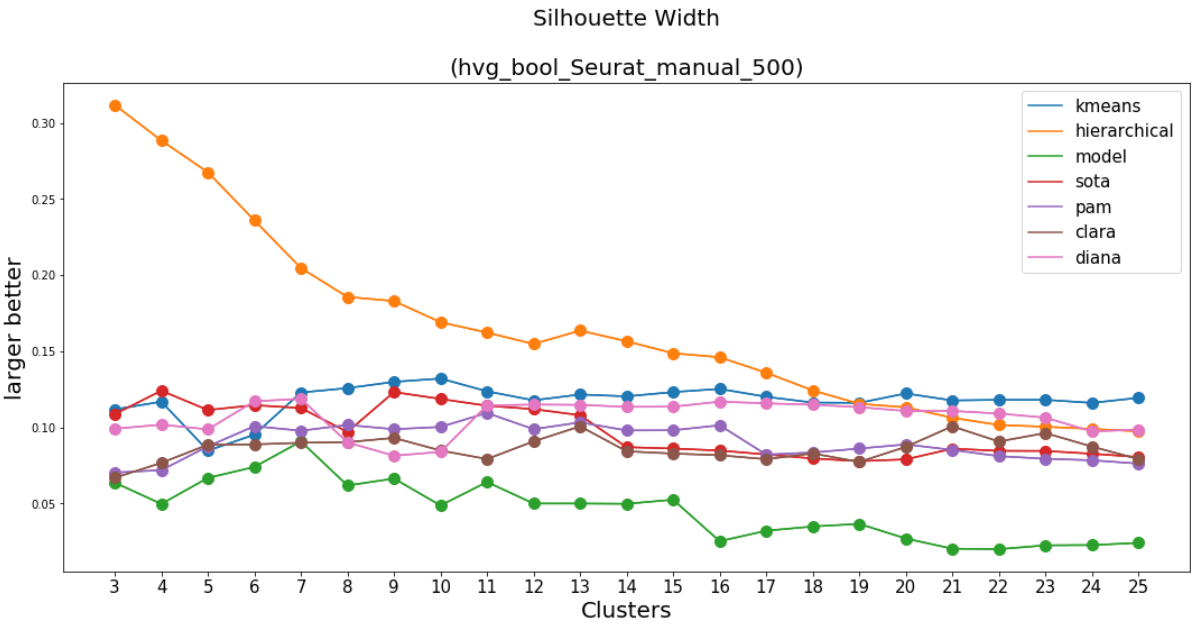


Figure 23. Silhouette Width, Seurat pre-processing and dispersion, 500 genes

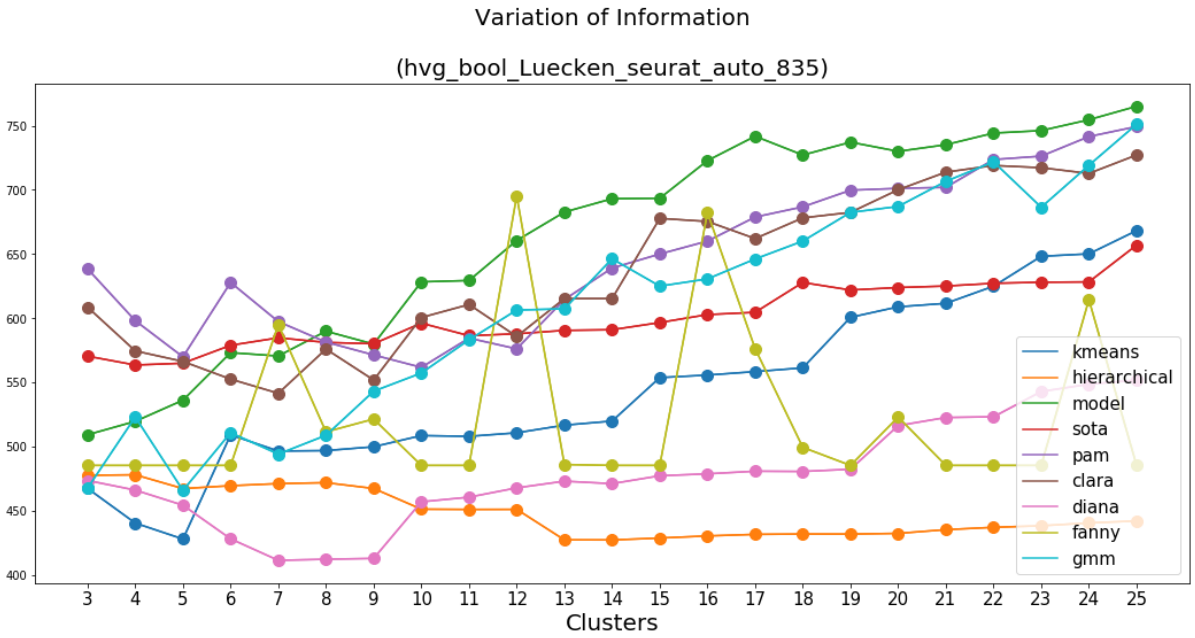


Figure 24. Variation of Information, Luecken pre-processing, Seurat dispersions, 835 genes

The labels of the two Louvain clusterings were compared to the once and twice clustered data used in the template examples. The Adjusted Rand Index and

Normalized Mutual Information (both use percentage as the unit of measurement) were 15-25% and 40-48% respectively (figure 25). The two new clusterings were more similar to each other in both measures than were the two Louvain clusterings (Figures 26 and 27).

```
Louvain Luecken r0.5 contains 10 different clusters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
With respect to the sub-clustered data it has:
        Adjusted Rand  0.237
        Normalized Mutual Information  0.465

With respect to the once-clustered data it has:
        Adjusted Rand  0.195
        Normalized Mutual Information  0.403



Louvain Luecken r1 contains 16 different clusters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
With respect to the sub-clustered data it has:
        Adjusted Rand  0.189
        Normalized Mutual Information  0.48

With respect to the once-clustered data it has:
        Adjusted Rand  0.154
        Normalized Mutual Information  0.421
```

Figure 25. Luecken pre-processing, Seurat dispersion, 3200 genes, k-means, 7 clusters for the once-clustered data; Luecken pre-processing, Seurat dispersion, 3200 genes, k-means, 44 clusters for the sub-clustered data.

```
Louvain Luecken r0.5 contains 10 different clusters
Louvain Luecken r1 contains 16 different clusters
        Adjusted Rand  0.392
        Normalized Mutual Information  0.643
```

Figure 26 Comparison between Louvain clusterings at different resolutions.

```
The once- (7 clusters) and twice- (44.0 clusters) clustered data
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       Adjusted Rand  0.791
       Normalized Mutual Information  0.866
```

Figure 27 Comparison both Luecken pre-processing, Seurat dispersion, 3200 genes, k- means, 7 clusters; same but 44 clusters

All of the following models used k-means with 7 clusters, but with different pre-processing, dispersion calculation method, and gene selection number. Regardless of the input, the Adjusted Rand and Normalized Mutual Information scores between the Louvain clustering and the new implementations of this report were consistently in the 15-25% and 35-50% ranges (Figures 27, 28 and 29). Despite this trend of similarities with the Louvain clustering, the similarities between the new implementations themselves varied dramatically, 6.2% to 54.5% for the Rand Index, and 10.6% to 53.9% for the Normalized Mutual Information (Figures 30, 31 and 32).

```
Louvain Luecken r0.5 contains 10 different clusters
hvg_bool_Luecken_all
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Normalized Mutual Information  0.373
        Adjusted Rand  0.139


Louvain Luecken r1 contains 16 different clusters
hvg_bool_Luecken_all
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Normalized Mutual Information  0.431
        Adjusted Rand  0.139
```

Figure 28. Comparison between Luecken pre-processing, all genes, k-means, 7 clusters and Louvain clustering.

```
Louvain Luecken r0.5 contains 10 different clusters
Cluster_labels_df_hvg_bool_Luecken_seurat_auto_835_obs_2000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Normalized Mutual Information  0.373
        Adjusted Rand  0.139


Louvain Luecken r1 contains 16 different clusters
Cluster_labels_df_hvg_bool_Luecken_seurat_auto_835_obs_2000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Normalized Mutual Information  0.431
        Adjusted Rand  0.139
```

Figure 29. Comparison between Luecken pre-processing, Seurat dispersion, 835 genes, k-means, 7 clusters and Louvain clustering.

```
Louvain Luecken r0.5 contains 10 different clusters
autoencoder_Luecken_pp_512
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Normalized Mutual Information  0.373
        Adjusted Rand  0.139


Louvain Luecken r1 contains 16 different clusters
autoencoder_Luecken_pp_512
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Normalized Mutual Information  0.431
        Adjusted Rand  0.139
```

Figure 30. Comparison between Luecken pre-processing, auto-encoded, 515 vector space, k-means, 7 clusters and Louvain clustering.

```
K-means, 7 clusters
bool_Luecken_seurat_auto_835_obs_2000 against Luecken_pp_Seurat_hvg_manual_3200
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       Adjusted Rand  0.545
       Normalized Mutual Information  0.636
```

Figure 31. Comparison between Luecken pre-processing, Seurat dispersion, 835 genes, k-means, 7 clusters and Luecken pre-processing, Seurat dispersion, 3200 genes, k-means, 7 clusters.

```
K-means, 7 clusters
autoencoder_Luecken_pp_512 against Luecken_pp_Seurat_hvg_manual_3200
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Adjusted Rand  0.062
        Normalized Mutual Information  0.106
```

Figure 32. Comparison between Luecken pre-processing, auto-encoded, 515 vector space, k-means, 7 clusters and Luecken pre-processing, Seurat dispersion, 3200 genes, k-means, 7 clusters.

```
K-means, 7 clusters
hvg_bool_Luecken_all against Luecken_pp_Seurat_hvg_manual_3200
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Adjusted Rand  0.229
        Normalized Mutual Information  0.539
```

Figure 33. Comparison between Luecken pre-processing, all genes, k-means, 7 clusters and Luecken pre-processing, Seurat dispersion, 3200 genes, k-means, 7 clusters.

5.0 Conclusion and Recommendations

The common approach to single-cell RNA-sequencing is too narrow. One pre-processing technique with one gene-selection technique with one dimensionality reduction technique with one clustering technique fails to reveal all of the information available in a data set. It may be the case that there are best individual choices for each of these tools. However, the mathematical analysis of the outcomes of different choices shows that there are statistically sound inferences from some of the other choices for these tools.

The five software templates develop herein can be easily inserted into existing workflows for practitioners. This will yield more insight into the structure and behaviour of biological processes at a cellular level, advance knowledge generally, as well as help with diagnosis and treatments for medical conditions.

REFERENCES

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). *Fast unfolding of communities in large networks.* Journal of statistical mechanics: theory and experiment, 2008(10), P10008.

G. Brock, V. Pihur, S. Datta, and S. Datta. clValid: *An R package for cluster validation*. Journal of Statistical Software, 25(4), March 2008. URL http:// www.jstatsoft.org/v25/i04.

Cancer Australia, Australian Government. (2019). *Cancer in Australia Statistics*. Retrieved from https://canceraustralia.gov.au/affected-cancer/what-cancer/cancer-australia-statistics

Chung, W., Eum, H. H., Lee, H. O., Lee, K. M., Lee, H. B., Kim, K. T., ... & Kan, Z. (2017). *Single-cell RNA-seq enables comprehensive tumour and immune cell profiling in primary breast cancer*. Nature communications, 8, 15081.

Ding, J., Condon, A., & Shah, S. P. (2018). *Interpretable dimensionality reduction of single cell transcriptome data with deep generative models*. Nature communications, 9(1), 2002.

Gates, A. J., & Ahn, Y. Y. (2017). *The impact of random models on clustering similarity.* The Journal of Machine Learning Research, 18(1), 3049-3076.

Haque, A., Engel, J., Teichmann, S. A., & Lönnberg, T. (2017). *A practical guide to single cell RNA-sequencing for biomedical research and clinical applications*. Genome medicine, 9(1), 75.

Kiselev, V. Y., Andrews, T. S., & Hemberg, M. (2019). *Challenges in unsupervised clustering of single-cell RNA-seq data*. Nature Reviews Genetics, 1.

Korunsky I. et al. (2018). *Fast, sensitive, and flexible integration of single cell data with Harmony*. Preprint at bioRxiv https://doi.org/10.1101/461954.

Lafzi, A., Moutinho, C., Picelli, S., & Heyn, H. (2018). *Tutorial: guidelines for the experimental design of single-cell RNA sequencing studies.* Nature protocols, 13(12), 2742-2757.

Luecken, M. D., & Theis, F. J. (2019). *Current best practices in single−cell RNA−seq analysis: a tutorial*. Molecular systems biology, 15(6).

Menden, K., Marouf, M., Dalmia, A., Heutink, P., & Bonn, S. (2019). *Deep-learning-based cell composition analysis from tissue expression profiles*. bioRxiv, 659227.

Moni, A. (2019). *Central Dogma*. IFN702 Project 2

Ntranos, V., Kamath, G. M., Zhang, J. M., Pachter, L., & David, N. T. (2016). *Fast and accurate single-cell RNA-seq analysis by clustering of transcript-compatibility counts*. Genome biology, 17(1), 112.

Ronan, T., Qi, Z., & Naegle, K. M. (2016). *Avoiding common pitfalls when clustering biological data.* Sci. Signal., 9(432), re6-re6.

Shen, L., Zhang, J., Lee, H., Batista, M. T., & Johnston, S. A. (2019). *RNA transcription and Splicing errors as a Source of cancer frameshift neoantigens for Vaccines*. Scientific reports, 9(1), 1-13.

Stoeckius, M., Zheng, S., Houck-Loomis, B., Hao, S., Yeung, B. Z., Mauck, W. M., ... & Satija, R. (2018). *Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics.* Genome biology, 19(1), 224.

Stuart, T., & Satija, R. (2019). *Integrative single-cell analysis*. Nature Reviews Genetics, 1.

Zhang, L., & Zhu, Z. (2019). *Unsupervised Feature Learning for Point Cloud by Contrasting and Clustering With Graph Convolutional NeuralNetwork*. arXiv preprint arXiv:1904.12359.

Zheng, S., Papalexi, E., Butler, A., Stephenson, W., & Satija, R. (2018). *Molecular transitions in early progenitors during human cord blood hematopoiesis*. Molecular systems biology, 14(3).