

# Clonal reconstruction of cancer from single cell expression based CNV inferences

by  
Kiran Sethi

Under the supervision of  
Dr. Debarka Sengupta

Submitted in partial fulfillment of the  
requirements for the degree of Master of  
Technology, Computational Biology



Center for Computational Biology Indraprastha  
Institute of Information Technology - Delhi  
August, 2022

# Certificate

This is to certify that the thesis titled “*Clonal reconstruction of cancer from single cell expression based CNV inferences*” being submitted by **Kiran Sethi** to the Indraprastha Institute of Information Technology Delhi, for the award of the Masters of Technology, is an original research work carried out by her under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or full to any other university or institute for the award of any degree/diploma.

August,2022

Dr Debarka Sengupta

Department of Computational Biology  
Indraprastha Institute of Information Technology Delhi  
New Delhi 110 020

# Acknowledgements

I would like to express my sincere gratitude and respect towards Prof. Debarka Sen-gupta from Indraprastha Institute of Information Technology, Delhi for being my super-visor and for giving me an opportunity to work on the wonderful topic and guiding me throughout. I would like to thank my parents and siblings for their continuous motiva-tion. Special thanks to my teammates Dinesh and Rishab for showing team spirit, my Ph.D. mentors Namrata Bhattacharya and Sarita Poonia for constantly providing their valuable feedback and guidance, and all my dear batchmates for always cheering me up. I also thank all the faculty members and staff of the Department of Computational Biology and IIIT Delhi for always helping us throughout our college journey. Also, I would like to thank Mr.Adarsh from the IT department, for his continuous support and help in providing access to the college IT infrastructure and helping us with the webserver deployment work.

# Abstract

Human cancers are composed of cells with varying genotypes, epigenetic states, and gene expression profiles. Intra and inter-tumor clonal heterogeneity is now recognized as one of the biggest drawbacks for therapeutic advancements in medical oncology. The clinical therapy of the condition is hampered by such an absurd degree of variability. For instance, a niche population may become tolerant and continue to develop and multiply when the majority of cancer cells die as a result of the toxicity brought on by a particular anti-cancer treatment. Thus it becomes crucial to find which of the clones has ancestry ties to the tolerant clone. This can be achieved by knowing the clonal phylogeny within the tumor. In this study, we tried to model the clonal phylogeny in cancer by using the single cell RNAseq data.

Single-cell RNA sequencing (scRNA-seq) is an emerging technology for profiling the gene expression of thousands of cells at single-cell resolution. This level of throughput analysis enables researchers to understand at the single-cell level what genes are expressed, in what quantities, and how they differ across thousands of cells within a heterogeneous sample. It can reveal complex and rare cell populations, uncover regulatory relationships between genes, and track the trajectories of distinct cell lineages in development which aims at understanding how a single-celled embryo gives rise to various cell types that are organized into complex tissue and organs.

However, the analysis comes up with a set of its own challenges like batch effects between datasets, limited availability of computational resources, and sharing restrictions on raw data. Recently, utilizing large-scale reference datasets to gain knowledge and then transferring it to smaller query datasets has become common in order to solve the above-mentioned problems. This concept is commonly known as Transfer Learning. In the second part of our study, we developed tranSCend which is a web server that hosts different pre-trained models accessible through a user-friendly interface to carry out different single cell analysis tasks. These tasks include data harmonizing, batch effect correction, normalization, visualization, clustering, cell-type classification, and differential gene expression analysis.

# Contents

<b>1</b>	<b>Modelling of intra-tumour clonal phylogeny using single cell RNAseq data</b>	<b>6</b>
1.1	Introduction . . . . .	6
1.2	InferCNV . . . . .	7
1.2.1	Input to inferCNV . . . . .	7
1.2.2	Generating CNV Profiles patient wise . . . . .	8
1.3	Data Collection . . . . .	9
1.4	Algorithm . . . . .	9
1.5	Result . . . . .	14
<b>2</b>	<b>Transcend - Webserver</b>	<b>16</b>
2.1	Introduction . . . . .	16
2.2	Data Collection and Pre-procesing . . . . .	17
2.2.1	Data Collection . . . . .	17
2.2.2	Data Pre-processing . . . . .	18
2.3	Tools . . . . .	19
2.4	Implementation of the Webserver . . . . .	20
2.4.1	Overview . . . . .	20
2.4.2	System Design . . . . .	20
2.4.3	Frontend . . . . .	21
2.4.4	Backend . . . . .	22
2.4.5	Deployment . . . . .	24
<b>3</b>	<b>Conclusion &amp; Future Scope</b>	<b>25</b>
3.1	Conclusion . . . . .	25
3.2	Future Scope . . . . .	25

# List of Figures

1.1	InferCNV Profile of BCH836 from pediatric midline gliomas tumor . . .	8
1.2	Scatter plot for BCH836 obtained after hierarchical clustering . . . . .	10
1.3	Signal difference A-B and B-A for pair(cluster 0,cluster 1) . . . . .	11
1.4	Constructing Confidence limits using Monte Carlo Randomization . . . .	12
1.5	A-B and B-A after low pass filter, BCH836 . . . . .	12
1.6	A-B is valid taking cluster 0 and cluster 1, BCH836 . . . . .	13
1.7	B-A is invalid taking cluster 0 and cluster 1, BCH836 . . . . .	13
1.8	Copy Number Variation Profile for BCH836 . . . . .	14
1.9	clonal phylogeny for BCH836 . . . . .	14
1.10	infercnv output for cy79 . . . . .	15
2.1	Snapshot of Model Browser . . . . .	21
2.2	Snapshot of Model Card . . . . .	22
2.3	Snapshot of Model Upload . . . . .	23

# Chapter 1

## Modelling of intra-tumour clonal phylogeny using single cell RNAseq data

### 1.1 Introduction

In cancer, the clonal composition varies within and between tumors. Next Generation Sequencing (NGS) technologies rapidly revolutionized the field of cancer genomics. In the recent few years, some insightful studies have revealed remarkable transcriptional heterogeneity within and between tumors by analyzing single-cell RNAseq data of hundreds of cells [1] [2]. Patel et. al. has shown that transcriptional heterogeneity results from clonal heterogeneity by inferring Copy Number Variation profiles from single cell RNAseq data. In many cancers, such as high-grade serous ovarian cancer (HGSOC), tumor heterogeneity is not reflected in point mutations but in copy-number profiles [3]. A copy number variation (CNV) is when the number of copies of a particular gene varies from one individual to the next. It is a type of structural variation where we have a stretch of DNA, which is duplicated in some people, and sometimes even triplicated or quadruplicated. And so when we look at that chromosomal region, we will see a variation in the number of copies in normal people.

Different cell types have different abilities in cancer. Sometimes a niche population becomes tolerant to drug toxicity and resumes proliferation [6]. Also, not all populations have the abilities of stem cells. Of note, EMT occurs only in a small number of cells. While it is absolutely important to appreciate heterogeneity in cancer, it is equally important to know how the heterogeneity emerges or how clones evolve. In the current

study, a bioinformatic approach is described for achieving the same.

## 1.2 InferCNV

InferCNV [1] is used to explore tumor single cell RNA-Seq data to identify evidence for somatic large-scale chromosomal copy number alterations, such as gains or deletions of entire chromosomes or large segments of chromosomes. This is done by exploring expression intensity of genes across positions of tumor genome in comparison to a set of reference 'normal' cells. A heatmap is generated illustrating the relative expression levels throughout each chromosome which parts of the tumour genome are over-expressed or under-expressed in comparison to those of normal cells. InferCNV gives users access to a variety of residual expression filters so they can experiment with reducing noise and more fully exposing the signal supporting CNA. Additionally, inferCNV offers strategies for predicting CNA locations and defining cell clusters in accordance with heterogeneity trends. InferCNV works based on the following approach:

- Genes which are not expressed in at least a predetermined number of cells are thrown away.
- Cells which do not express at least a predetermined number of the selected genes are discarded.
- Genes are first sorted according to their position on the human genome.
- Expressions are then centred along libraries and then genes.
- Moving average of expression is computed along the genes, sorted by their respective genomic locations. The vector with rolling average values is considered as the pseudo CNV profile of a single cell.

### 1.2.1 Input to inferCNV

- **Raw Counts Matrix** of single-cell RNA-Seq expression which can be generated using any conventional single cell transcriptome quantification pipeline, giving a matrix of genes vs. cells in (row vs column format) containing assigned read counts.



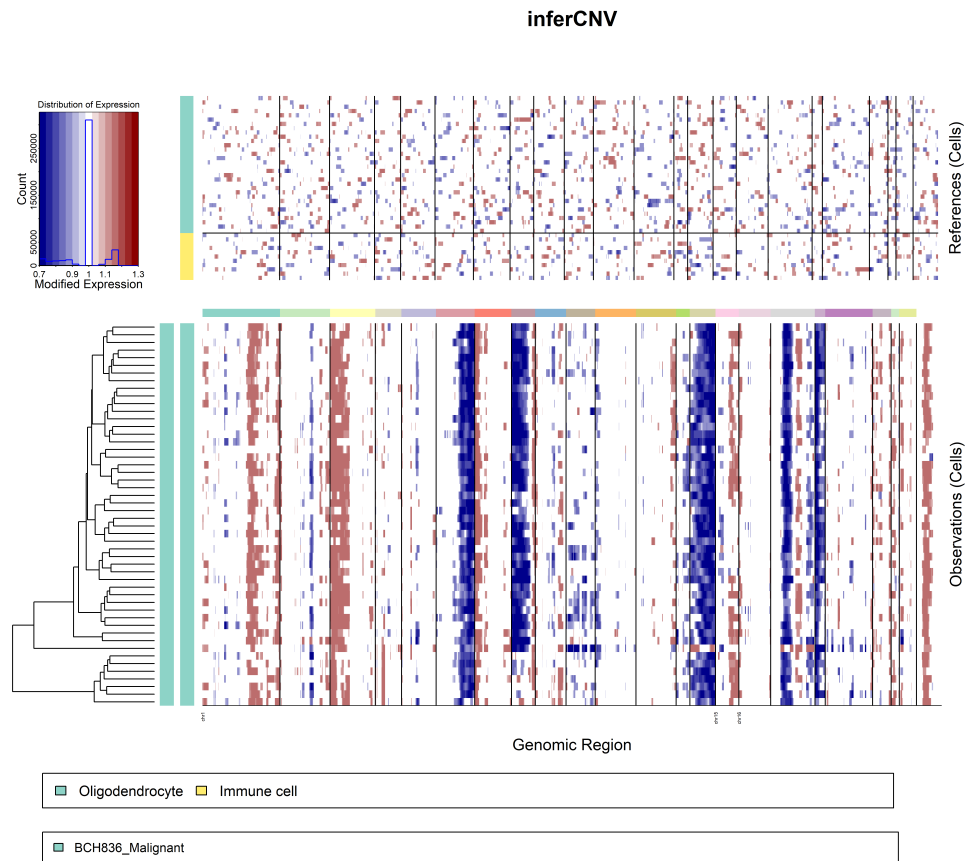


Figure 1.1: InferCNV Profile of BCH836 from pediatric midline gliomas tumor

- **Annotations file** that indicates which cells are tumor vs. normal having two columns where the first column represents the cell name and the second column represents known cell type along with the patient id.
- **Gene/Chromosome positions file** that provides the chromosomal location for each gene. The format is tab-delimited and has no column header, simply providing the gene name, chromosome, and gene span.

## 1.2.2 Generating CNV Profiles patient wise

Most of the time the publicly available data for the tumor contains the cells of different patients taken over different biopsies over a different period of time. The raw count

matrix that we obtained consists of the malignant and non-malignant cells of different patients. To obtain the Copy Number Profiles per patient, we modified the input annotation file and kept only those malignant cells that correspond to the patient we desired. In this way, we obtained the patient-wise CNV profile of the tumor cells.

### 1.3 Data Collection

Data for different tumors were derived from the following sources. The Algorithm for construction of clonal phylogeny was tested on the Copy Number profiles generated from these tumors per patient using inferCNV.

- **Glioblastoma** derived from ([https://portals.broadinstitute.org/single\\_cell/study/glioblastoma-intra-tumor-heterogeneity](https://portals.broadinstitute.org/single_cell/study/glioblastoma-intra-tumor-heterogeneity)) [4]
- **Melanoma** derived from ([https://portals.broadinstitute.org/single\\_cell/study/melanoma-intra-tumor-heterogeneity](https://portals.broadinstitute.org/single_cell/study/melanoma-intra-tumor-heterogeneity)) [5]
- **Oligodendroglioma** derived from ([https://portals.broadinstitute.org/single\\_cell/study/oligodendroglioma-intra-tumor-heterogeneity](https://portals.broadinstitute.org/single_cell/study/oligodendroglioma-intra-tumor-heterogeneity)) [6]
- **Pediatric midline gliomas** derived from ([https://portals.broadinstitute.org/single\\_cell/study/single-cell-analysis-in-pediatric-midline-gliomas-with-histone-h3k27m-mutation](https://portals.broadinstitute.org/single_cell/study/single-cell-analysis-in-pediatric-midline-gliomas-with-histone-h3k27m-mutation)) [7]

### 1.4 Algorithm

- **Step 1. Create the Annotation file patient wise**  
To work on the tumor patient wise , we need to generate the Copy Number Variation Profile of the patients. The public dataset we used contains the gene expression data of the tumour for multiple patients. So, To generate the CNV per patient , we modified the annotation file that's used as the input to the inferCNV. The annotation file is the tab limited text file having the following structure containing both the malignant cells and the reference cells:

BCH836-P01-A03  
BCH836-P01-C12

BCH836-Malignant  
Immune cell

The first column indicates cell name and the second column indicates the cell type. Patient id can be observed from the cell name. Malignant cells of only patient of interest is kept in the annotation file for example, BCH836 to generate patient wise Copy Number Variation Profile.

- **Step 2. Generate Copy Number profile using InferCNV:** Heatmap for Copy Number Profile of the patient BCH836 generated using inferCNV shown in 1.1. Heatmap drawn on the lower section, named as Observations corresponds to residual expression of malignant cells while in the upper section, the heatmap named References corresponds to the non malignant cells.
- **Step 3. Clustering the InferCNV output expression matrix:** Following the identification of the cell-specific CNV profiles, aggregation was performed using a clustering technique. Correlation between CNV profiles can be utilised as a distance metric. To obtain clusters, we used ward linkage hierarchical clustering. After that, Each individual clone corresponds to a distinct cluster. Figure 1.2 shows the scatter plot obtained by hierarchical clustering.

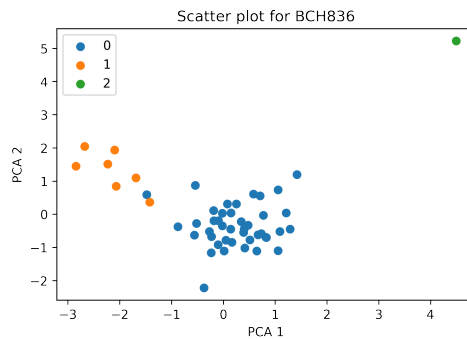


Figure 1.2: Scatter plot for BCH836 obtained after hierarchical clustering

- **Step 4. For each cluster, generate the average of the expression:** The average vector of the expression corresponding to each cluster will serve as the expression for the clone. For our further procedure, we considered this vector as the representation of the clone.

- **Step 5. Shifting mean to the origin:** The distribution of the expression matrix is normal like and is centered around mean. For example in the BCH836 CNV profile as shown in 1.1 , the expression is centered around 1. We shifted the mean to the origin by subtracting 1 from each value across the genomic region for all the samples.
- **Step 6. nC2 Combinations , n= number of clusters:** To establish clonal phylogeny, relationship if exists between the clones need to be find out .For this purpose ,each combination pair from the clusters are considered.Relationship is checked for total nC2 combinations where n is the number of clusters.
- **Step 7. Generate both A-B and B-A:** Take all the possible pair combinations of the average expression of the clusters .Each combination will have signals corresponding to two clusters which will be taken as A and B. For each combination , A-B and B-A both are generated. For ex: A= cluster 0 , B= cluster 1 for BCH836

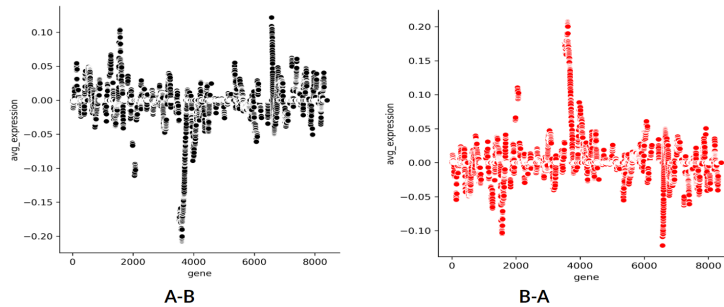


Figure 1.3: Signal difference A-B and B-A for pair(cluster 0,cluster 1)

- **Step 8. Noise filter:** Signals often have noise associated with them.To eliminate the noise in the signal vector corresponding to the clusters , we used **Monte Carlo Randomization**.Using Randomization we can construct confidence limits.Monte Carlo randomization quantitatively evaluates observed data and test statistics. We considered the extreme 5 percent as the significant signal difference . The 95 percent confidence interval value is treated as noise. The detailed procedure is as following:  
i) Randomly sampled 2 signals ,let's say A and B from the complete inferCNV output expression matrix at a time and consider their difference i.e A-B.This process is done iteratively till certain times.

ii) According to the Central Limit Theorem ,the density plot of A-B sampled over the dataset will have Gaussian like distribution.

iii)All values lying between the 95 percent interval is considered as noise . This is eliminated from our signal A-B and B-A as discussed in the previous section.The 95 percent confidence interval lies in the range-  $[\mu - 1.96*\sigma, \mu + 1.96*\sigma]$

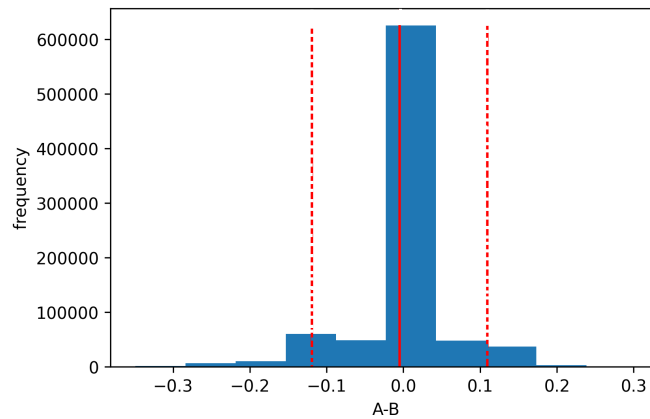


Figure 1.4: Constructing Confidence limits using Monte Carlo Randomization

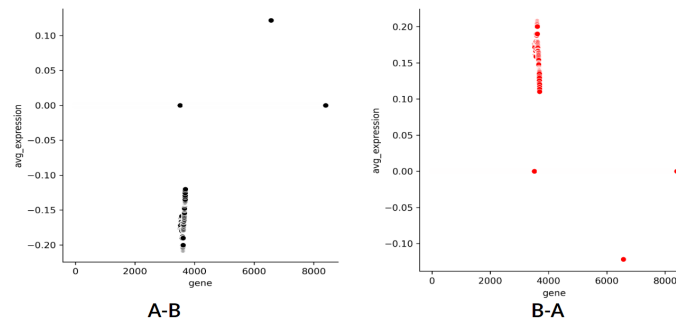


Figure 1.5: A-B and B-A after low pass filter, BCH836

● **Step 9. Constructing Phylogeny:**

For difference signal in [A-B , B-A]

- 1.Count the number of spikes.
- 2.Every spike should have direction similar to that of any parent.
- 3.The absolute magnitude of the spike of difference signal should be less than the parent signal.

4.If the condition hold true for every spikes in the difference signal ,then difference signal is valid otherwise ,no direct relation.

if A-B is valid , relation will be  $B \rightarrow A$  if B-A is valid ,relation will be  $A \rightarrow B$

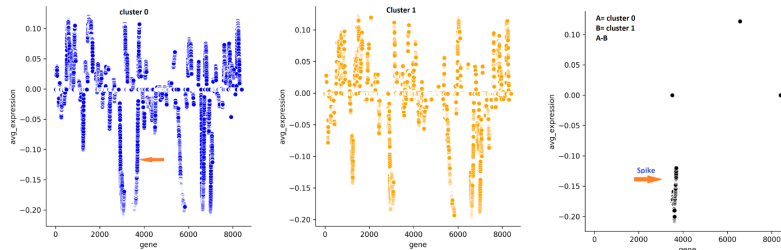


Figure 1.6: A-B is valid taking cluster 0 and cluster 1, BCH836

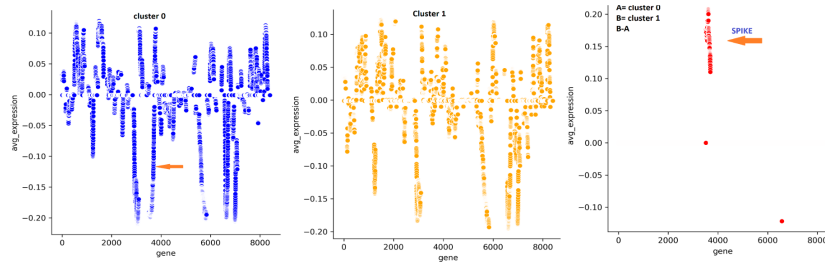


Figure 1.7: B-A is invalid taking cluster 0 and cluster 1, BCH836

As per figure 1.6 ,A-B is valid .Its spike coincides with the spike in A i.e cluster 0 , also magnitude is less than the parent.

As per figure 1.7 ,B-A is invalid.All the spikes should coincide with any parent and absolute value should be less than the parent . This does not hold true in B-A.

- **Step 10. Spanning Tree:** The relationships between all the possible clusters taken two at a time are evaluated.All the transitive relationships are eliminated. Using minimum spanning tree, clonal phylogeny is constructed.

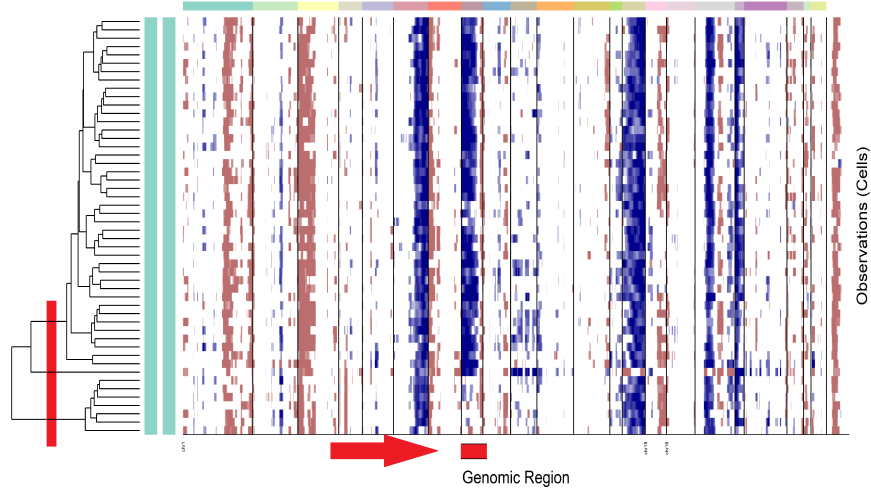


Figure 1.8: Copy Number Variation Profile for BCH836

**A= cluster 0**  
**B= cluster 1**  
**C=cluster 2**  
**N =missing clone**

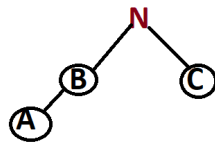


Figure 1.9: clonal phylogeny for BCH836

## 1.5 Result

For pediatric midline gliomas ,patient BCH836 the Copy Number Variation profile is shown in the figure 1.1 and our resulting clonal phylogeny is shown in figure 1.9.

Considering another example, for Melanoma tumor and patient CY79 the Copy Number Variation Profile is shown in the figure 1.10

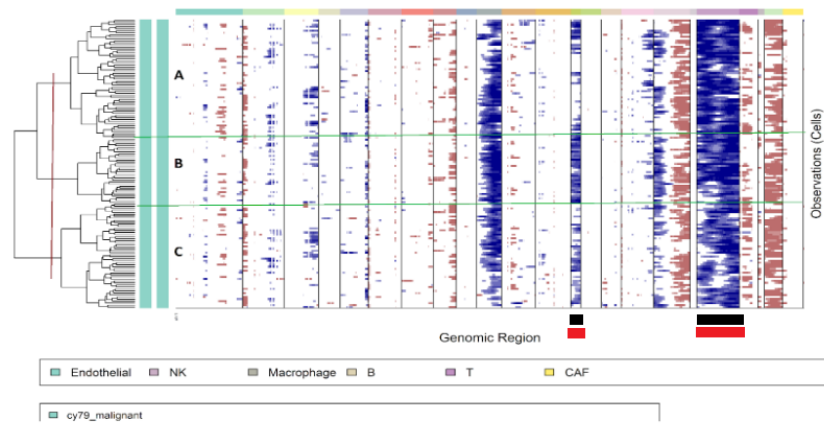


Figure 1.10: infercnv output for cy79

The clonal phylogeny for Melanoma tumor for the patient CY79 came out to be  $C \rightarrow A \rightarrow B$ .

In this way, we tried to model the intra tumor heterogeneity by constructing clonal phylogeny using single cell RNA sequencing data.



# Chapter 2

## Transcend - Webserver

### 2.1 Introduction

Large-size cell atlases are now available for single-cell analysis thanks to developments in single-cell sequencing methods. However, using these reference atlases has always been difficult because of the batch effect that occurs when various technologies are used to sequence different datasets. The technological variation in the data brought on by the batch effect obscures biological variation and produces meaningless results. To solve the problem, various machine learning approach has been suggested.

However, building models from scratch for single cell analysis pipelines requires a lot of data and computing. Another difficulty is ensuring generalization to new, unobserved data, particularly in environments where target data is scarce. Utilizing pre-trained models with sufficient domain knowledge that have previously been trained and then using fresh data to improve the current models is a straightforward but efficient solution to both issues. Transfer learning is the name given to this strategy. Tasks like batch correction, clustering, imputation, and others can all be incorporated into the transfer learning framework and profit from it (in terms of generalization performance).

Due to the availability of extensive reference cell atlases that may be used to solve tasks particular to target data, where target data has comparatively few data samples, the transfer learning (TL) technique has also become increasingly popular in the processing of single-cell RNA-seq data. For instance, scANVI, trVAE, scRNA, ItClust, scETM , etc have shown the potential of transfer learning for single-cell sequencing, by showing improved performance on clustering results.

Analyzing the emerging trend of Transfer Learning in single-cell RNA sequencing, a

need is felt for a platform that provides both pre-trained models with associated meta-data as well as examples demonstrating complete end-to-end use of these models. With the goal of curating a large collection of cutting-edge, already-trained transfer learning models for single-cell analysis that are created by and accessible to the community, we created an open-source web server called tranSCend. tranSCend hosts the pre-trained models by acting as a model repository that have been trained on sizable reference datasets and can be fine-tuned according to the requirements of query datasets. The whole code for fine-tuning these models, including references to the datasets used, is supplied in an end-to-end Python notebook.

## 2.2 Data Collection and Pre-processing

### 2.2.1 Data Collection

For deep learning techniques to be effective, large volumes of data are often needed. A sizable number of single cell datasets from various organ systems and species are publicly accessible online. The focus of this work is on single cell data from mouse and human. We used data of the organs like kidney, pancreas, blood, spleen, lung epithelial, prostate and colon. Following is the list of prominent sources:

- **Covid 19 cell atlas** (<https://www.covid19cellatlas.org/>)

This website contains a wide variety of single cell datasets from various sources. In most cases, both the dataset and the associated literature are made available. Datasets from both healthy as well as patient donors are available.

- **Gene Expression Omnibus** (<https://www.ncbi.nlm.nih.gov/geo/>)

This is a well-known public functional genomics data repository. Datasets can be accessed using their GSE accession number and freely downloaded using command line utilities like wget.

- **10X Genomics** (<https://www.10xgenomics.com/>)

With the use of chromium technology, it is a commonly utilized sequencing technique that can sequence millions of cells. Whether it be scRNA-seq, scATAC-seq, immunological profiling, spatial gene expression, or target gene expression data, this technique encompasses various modalities of single-cell data. The data format 10x genomics is also compatible with packages like scanpy (single-cell analysis python package )

## 2.2.2 Data Pre-processing

Generally, biases, artifacts, and other sources of unwanted variation are present in the data. This requires substantial time and effort to be spent on pre-processing and normalization. We have followed a common pipeline of pre-processing for both the reference and the query datasets. This pipeline comprises of the following steps:

- **Data Reading**

With the introduction of Seurat [8] package in R, scRNA count matrix could be read as a Seurat object. The object serves as a container that contains both data (like the count matrix) and analysis (like PCA, or clustering results) for a single-cell dataset. This improved the analysis as it became easy to extract required data from the object. Later on, many such packages get developed which would enable us to read the count matrix in the form of object. Scanpy is one such library in python which tries to replicate the functionalities of Seurat. We chose Scanpy [9] for our work as python is one of the most versatile language and many pre-trained models are developed in python itself.

Scanpy python package reads in the form of annotated data object (or ann data object). This data object contains a mix of pandas dataframes and numpy arrays. The matrix is stored in a numpy ndarray format, where the dimension is  $n$  (no. of observations) \*  $d$  (no. of variables). Observations refer to the cells (cell barcodes) and variables correspond to the genes (gene symbols/ gene ids). The annotations for the observations and variables are stored in the form of pandas series objects. Annotations typically contain data on cell type, batch/study, metrics for genes' mean and dispersion scores, information on highly variable genes, etc. The count matrix, genes file, and barcode file are typically read individually and then integrated into a single annotated data object.

- **Normalizing and Log transforming the data**

Due to sequencing depth, the count matrix has a different range of values for features. Normalization is performed for this reason. CPM(counts per million) followed by log transformation is performed on the datasets.

- **Highly variables genes selection**

scRNA datasets have a large number of features(genes). Thus, it's important to reduce the number of features for the improving efficiency of training the model and also preventing the curse of dimensionality. We select only those genes for further analysis which are highly variable.

## 2.3 Tools

Transfer learning has been performed using different state-of-art architectures like **scVI** [10], **trVAE**[11], **scANVI** [12], **LATE**[13] etc across different datasets covering human and mouse species and major organs like pbmc,pancreas,kidney,colon,etc.

Workflows for scRNA-seq incorporate numerous traditional machine learning techniques. The normalisation of the data is the first step in the workflow, which also involves reducing the dimensions of the data for visualisation, using an ad-hoc algorithm to correct batch effects, clustering the data to identify cell states from the corrected latent space, and then using differential expression to match the clusters to recognised cell types. Given that each step in the workflow makes a specific assumption, **scVI** produced a unified model of assumptions for the entire pipeline. Consequently, a deep generative model called **scVI** is presented that takes on all of these tasks.

**Single-cell Annotation using Variational Inference (scANVI)**, is another method that extends scVI and provides a principled way to address the annotation problem probabilistically while leveraging any available label information.[10, 14]

Similarly, **LATE** is an autoencoder neural network that performs the Imputation of single-cell gene expression based on a set of deep learning algorithms to recover the true gene expression values.

**Single-cell Embedded Topic Model (scETM)**, a generative topic model composed of a linear decoder that makes use of matrix trifactorization and a neural network-based encoder. A set of highly interpretable gene embeddings, topic embeddings, and batch-effect linear intercepts are all simultaneously learned by the model from scRNA-seq data together with the encoder network parameters.

Other transfer learning techniques include **transformer variable auto encoding (trVAE)**, which address the challenge of transforming out-of-sample by regularizing the joint distribution across the categorical variables using maximum mean discrepancy (MMD) in the framework of a conditional variational autoencoder (CVAE). This produces a more compact representation of a distribution that displays high variance in the vanilla CVAE, which incentivizes learning of features across s and results in more accurate out-of-sample prediction. trVAE shows qualitatively improved predictions for cellular perturbation response to treatment and disease based on high-dimensional single cell gene expression data by providing better estimates for mean and variance

compared to other models and also handling multiple conditions.

## 2.4 Implementation of the Webserver

### 2.4.1 Overview

tranSCend is a scalable and easy-to-use platform for anyone to upload their state-of-art transfer learning models or existing models fine-tuned on new query data. This can be done using the upload section of the webserver. The model along with the necessary meta information can be uploaded easily. Once verified, this is available in the model browser and is accessible to all the users.

tranSCend reduces the user's overhead of searching for various models and methods for carrying out single-cell analysis by providing all the relevant models on a single platform, thus saving their time for training the models on reference datasets from scratch and minimizing their use of computational resources. Users can download the already existing pre-trained models in the Model Browser section of the webserver. All the meta information like model name, the dataset on which it is trained, number of cells in that particular data, contributor of the model, task performed by the model like classification, clustering, and imputation along with the downloadable model files (learned weights/parameters) and python notebook is present in the model browser in a tabular format.

Searching and filtering the table comes up with other features like grouping based on any meta information, sorting the models according to the number of downloads and likes, etc. Detailed information regarding each model is present on the model card associated with it. The jupyter notebooks corresponding to the models contain the end-to-end procedure for using the pre-trained models for fine-tuning them on the desired query data and then using the output for downstream tasks.

### 2.4.2 System Design

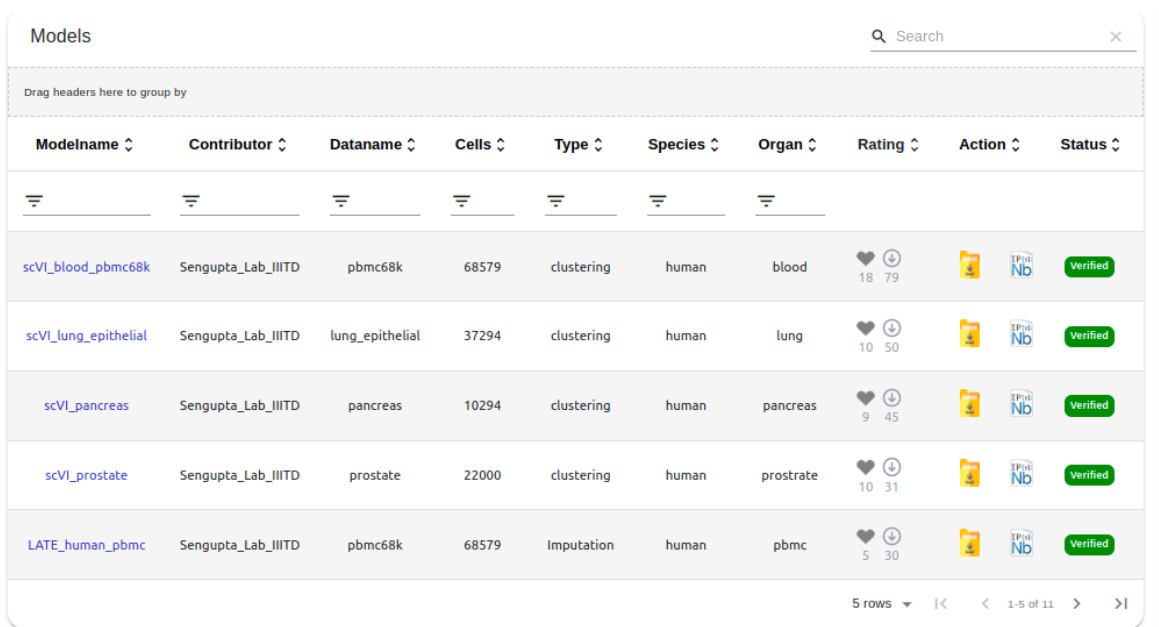
The frontend of tranSCend is a single-page React application written in JavaScript. The pre-trained models and their fine-tuning code along with the meta-information are stored on the server and the PostgreSQL database is used for the same. This data is retrieved or updated using the Rest API implemented in the Django Rest Framework.

## 2.4.3 Frontend

The application is designed such that its core functionalities can be easily accessed by the users. Its majorly divided into three components, namely Model Browser, Upload and Model Card.

- **Model Browser**

The pre-trained models are available in the **Model Browser** along with the downloadable python notebook for fine-tuning in a tabular format. The table also displays other information related to the pre-trained model like contributors, data upon which the model is trained, data specific information like the number of cells, species, organs, and verification status. The idea of verification status is to ensure greater transparency amongst the users. The rating status is maintained by the frequency of the model being downloaded or the model being liked.



Modelname	Contributor	Dataname	Cells	Type	Species	Organ	Rating	Action	Status
scVI_blood_pbmc68k	Sengupta_Lab_IITD	pbmc68k	68579	clustering	human	blood	18 79		Verified
scVI_lung_epithelial	Sengupta_Lab_IITD	lung_epithelial	37294	clustering	human	lung	10 50		Verified
scVI_pancreas	Sengupta_Lab_IITD	pancreas	10294	clustering	human	pancreas	9 45		Verified
scVI_prostate	Sengupta_Lab_IITD	prostate	22000	clustering	human	prostate	10 31		Verified
LATE_human_pbmc	Sengupta_Lab_IITD	pbmc68k	68579	Imputation	human	pbmc	5 30		Verified

Figure 2.1: Snapshot of Model Browser

- **Model Card** provides extensive information about the pre-trained models available on the Model Browser. This includes :
  - a) Model Information which consists of Model Name and Model Architecture.
  - b) Citation information for both the training dataset and pre-trained model.
  - c) Description of the model.

- d) Data Information which consists of information about the data on which the model is trained like the Data Source ,Species , Transcriptome Version , Reference and download link , Number of cells and Number of genes of the dataset.
- e) Jupyter notebook corresponding to the model having end-to-end python code for how to use.

**Model Information** ♥ like | 18

**Model Name :** scVI\_blood\_pbmc68k  
**Model Architecture :** Bayesian hierarchical model

**Citation**

**Model Cite:**  
 Lopez, R., Regier, J., Cole, M.B. *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* 15, 1053–1058 (2018)[<http://doi.org/10.1038/s41592-018-0229-2>]

**Dataset Cite:**  
 Fresh 68k PBMCs (Donor A), Single Cell Gene Expression Dataset by Cell Ranger 1.1.0, 10x Genomics, (2016, July 24) [<https://www.10xgenomics.com/resources/datasets/fresh-68-k-pbm-cs-donor-a-1-standard-1-1-0>]

**Description**

**scVI** is a generative model intended for the analysis of scRNAseq data. It explicitly models two chief nuisance factors in scRNAseq data i.e. library size and batch effects. It is based on a hierarchical bayesian model with conditional distributions specified by neural networks. In some detail, the observed expression  $x_{ng}$  of each gene  $g$  in cell  $n$  is modelled as a sample drawn from a zero-inflated negative binomial (ZINB) distribution  $p(x_{ng}|z_n, s_n, l_n)$ . Here  $s_n$  is the batch annotation of each cell(if available) and  $l_n$  and  $z_n$  are two additional unobserved random variables.  $l_n$  which is a one dimensional Gaussian RV that represents nuisance variation due to differences in capture efficiency and sequencing depth.  $z_n$  is a low dimensional vector of Gaussian that represents variation owing to biological differences between cells. Each cell is represented as a point in a low dimensional latent space. This representation can be used for clustering and subsequent visualization. A neural network is used to map the latent variables to the parameters of the ZINB distribution.

Figure 2.2: Snapshot of Model Card

- **Upload**

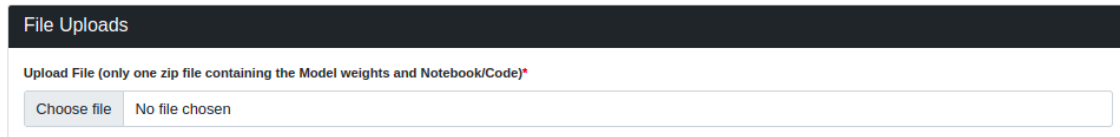
tranSCend is designed to be extensible by researchers. Anyone can upload their state-of-art architecture for community use in the **Upload** section of the application along with the meta information as well as jupyter notebook containing the instructions to use the pre-trained model.

#### 2.4.4 Backend

The backend is built using Django and Django REST framework is used for the APIs. Django offers an easy-to-use ORM (Object Relational Mapper) for database querying, which lets users avoid using raw SQL queries while performing database CRUD operations. Postgresql which provides secure databases has been used for the database. The REST (Representational State Transfer) APIs provide the way to transfer the data between the frontend and the database.

# Model Upload

Upload the pre-trained source models for transfer learning. You can submit your pre-trained model on a new or existing reference and share the information about the model and the data with other researchers.



File Uploads

Upload File (only one zip file containing the Model weights and Notebook/Code)\*

Choose file No file chosen

Figure 2.3: Snapshot of Model Upload

The **Database** consists of two tables namely the model and the user table linked using the contributor. A brief description of both the tables is mentioned below.

The first table contains information about the model.

- **ModelName:** model name for the specific methods used
- **Contributor:** the name of lab, person or organization who is responsible for the hosting of these models
- **DataName:** the dataset name on which the model is trained
- **Species:** the species of the dataset like human, mouse etc
- **Organ/ Tissue:** the organ/ tissue name on which model is trained like blood, liver, kidney and pancreas etc
- **Action:** This provides clickable buttons for downloading the model file and notebook file
- **Rating:** it provides the statistics for the number of times a model is downloaded and liked by the user
- **Number of cells:** number of cells used while training the model
- **Number of genes:** number of genes: number of genes used while training the model



- **Model reference/source:** the research paper / GitHub link referred for the paper, so that the model can easily be cited
- **Dataset reference/source:** dataset source paper or website for citation
- **Dataset download link:** For downloading the dataset
- **File upload(single zip file containing model and notebook):** This is a folder upload tab where a single zip file is uploaded containing the pretrained model file and associated code notebook used for training/ finetuning the code.

The second database table contains basic user information who has uploaded/ hosted the model like:

- **Name:** Name of the person having the issue/query/message
- **User Name:** name of the lab or organization the person is associated with
- **Email:** mailing address of the person
- **Occupation:** occupation of the person like student, faculty, researcher etc

Different **APIs** have been used in the application. These APIs has been developed using Django Rest Framework. The API function ranges from fetching the models from the model table in the Model Browser, posting the model and the information in the model table, updating the status of download and likes of the model and also to fetch trending 4 models for the homepage.

## 2.4.5 Deployment

Website is deployed on the IIIT Delhi server i.e 192.168.17.155. Frontend and Backend are hosted on the same server ; this is achieved by including the frontend as build folder in the Django backend. For the application's deployment, we use Gunicorn and Nginx. Gunicorn works closely with the Django backend to stop client requests from directly interacting with server code. The process uses the Nginx server, which collects all requests and sends them to Gunicorn so it can retrieve the necessary information.

The server is assigned a domain name and secured using LetsEncrypt.transCend can be accessed at <https://transcend.senguptalab.iiitd.edu.in/>.

# Chapter 3

## Conclusion & Future Scope

### 3.1 Conclusion

Understanding more about intra tumour heterogeneity can help us better comprehend cancer. It is one of the oncology field's challenges. Also ,with the advent of single cell genomics ,we have new insights into the fundamentals of biology. We tried to model the intra tumor heterogeneity using the scRNA data, going one step ahead of the already existing work which says that transcriptional heterogeneity is related to the copy number variation profile. We constructed the clonal phylogeny within the tumor trying to answer question like which of the clones has ancestry ties to the tolerant clone.

In the second part of our study,we developed open-source community-scale tran-SCend which is a web server that hosts different pre-trained models accessible through a user-friendly interface to carry out different single cell analysis tasks.The idea was to leverage the power of Transfer Learning in single cell genomics and providing the community a one stop solution to carry out single cell analysis.

### 3.2 Future Scope

While constructing clonal phylogeny ,signal based approach is followed.Counting number of spikes in the signal is the crucial step.However , currently only one way is explored for counting the number of spikes.For checking if spikes are occurring, first the noise is removed from the signal.Further,the pattern of long running zeroes having consecutive non zeroes values are identified.These patterns are termed as spikes.The current method give equal importance to all the spikes discovered this way.In future, we hope to explore the other ways in which spike could be explored where the spike magnitude

importance also comes up in the scenario.

tranSCend can be further extended and more pre-trained models can be hosted. Also ,currently the code to use or fine-tune the pre-trained models come up in jupyter notebook. In the future , Rest API could be provided to directly use the models.

# Bibliography

- [1] A. P. Patel, I. Tirosh, J. J. Trombetta, A. K. Shalek, S. M. Gillespie, H. Wakimoto, D. P. Cahill, B. V. Nahed, W. T. Curry, R. L. Martuza *et al.*, “Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma,” *Science*, vol. 344, no. 6190, pp. 1396–1401, 2014.
- [2] P. Dalerba, T. Kalisky, D. Sahoo, P. S. Rajendran, M. E. Rothenberg, A. A. Leyrat, S. Sim, J. Okamoto, D. M. Johnston, D. Qian *et al.*, “Single-cell dissection of transcriptional heterogeneity in human colon tumors,” *Nature biotechnology*, vol. 29, no. 12, pp. 1120–1127, 2011.
- [3] W. Jones, “Genomics and bioinformatics in biological discovery and pharmaceutical development,” pp. 105–142, 2020.
- [4] J. Peng, B.-F. Sun, C.-Y. Chen, J.-Y. Zhou, Y.-S. Chen, H. Chen, L. Liu, D. Huang, J. Jiang, G.-S. Cui *et al.*, “Single-cell rna-seq highlights intra-tumoral heterogeneity and malignant progression in pancreatic ductal adenocarcinoma,” *Cell research*, vol. 29, no. 9, pp. 725–738, 2019.
- [5] I. Tirosh, B. Izar, S. M. Prakadan, M. H. Wadsworth, D. Treacy, J. J. Trombetta, A. Rotem, C. Rodman, C. Lian, G. Murphy *et al.*, “Dissecting the multicellular ecosystem of metastatic melanoma by single-cell rna-seq,” *Science*, vol. 352, no. 6282, pp. 189–196, 2016.
- [6] I. Tirosh, A. S. Venteicher, C. Hebert, L. E. Escalante, A. P. Patel, K. Yizhak, J. M. Fisher, C. Rodman, C. Mount, M. G. Filbin *et al.*, “Single-cell rna-seq supports a developmental hierarchy in human oligodendroglioma,” *Nature*, vol. 539, no. 7628, pp. 309–313, 2016.
- [7] M. G. Filbin, I. Tirosh, V. Hovestadt, M. L. Shaw, L. E. Escalante, N. D. Mathewson, C. Neftel, N. Frank, K. Pelton, C. M. Hebert *et al.*, “Developmental and

- oncogenic programs in h3k27m gliomas dissected by single-cell rna-seq,” *Science*, vol. 360, no. 6386, pp. 331–335, 2018.
- [8] E. Z. Macosko, A. Basu, R. Satija, J. Nemesl, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck *et al.*, “Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets,” *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.
- [9] F. A. Wolf, P. Angerer, and F. J. Theis, “Scanpy: large-scale single-cell gene expression data analysis,” *Genome biology*, vol. 19, no. 1, pp. 1–5, 2018.
- [10] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, “Deep generative modeling for single-cell transcriptomics,” *Nature methods*, vol. 15, no. 12, pp. 1053–1058, 2018.
- [11] M. Lotfollahi, M. Naghipourfar, F. J. Theis, and F. A. Wolf, “Conditional out-of-sample generation for unpaired data using trvae,” *arXiv preprint arXiv:1910.01791*, 2019.
- [12] C. Xu, R. Lopez, E. Mehlman, J. Regier, M. I. Jordan, and N. Yosef, “Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models,” *Molecular systems biology*, vol. 17, no. 1, p. e9620, 2021.
- [13] M. Badsha, R. Li, B. Liu, Y. I. Li, M. Xian, N. E. Banovich, A. Q. Fu *et al.*, “Imputation of single-cell gene expression with an autoencoder neural network,” *Quantitative Biology*, vol. 8, no. 1, pp. 78–94, 2020.
- [14] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.