Calculating sample size for identifying cell subpopulation in single-cell RNA-seq experiments

Kyung In Kim*, Ahrim Youn, Mohan Bolisetty, A. Karolina Palucka, Joshy George*

The Jackson Laboratory for Genomic Medicine,

10 Discovery Drive, Farmington, Connecticut, USA

kyungin.kim@jax.org, josh.george@jax.org

SUMMARY

Single-cell RNA sequencing (scRNA-seq) is a rapidly developing technology for studying gene expression at the individual cell level and is often used to identify subpopulations of cells. Although the use of scRNA-seq is steadily increasing in basic and translational research, there is currently no statistical model for calculating the optimal number of cells for use in experiments that seek to identify cell subpopulations. Here, we have developed a statistical method ***ncells*** for calculating the number of cells required to detect a rare subpopulation in a homogeneous cell population for the given type I and II error. ***ncells*** defines power as the probability of separation of subpopulations which is calculated from three user-defined parameters: the proportion of rare subpopulation, proportion of up-regulated marker genes of the subpopulation, and levels of differential expression of the marker genes. We applied ***ncells*** to the scRNA-seq data on dendritic cells and monocytes isolated from healthy blood donor to show its efficacy in calculating the optimal number of cells in identifying a novel subpopulation.

*To whom correspondence should be addressed.

Key words: Sample size calculation; single-cell RNA-seq; Dropout

1. INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) technology (Tang *and others*, 2009; Ramsköld *and others*, 2012) has been rapidly applied to many biomedical research areas such as cancer (Patel *and others*, 2014), neurobiology (Tasic *and others*, 2016) and immunology (Villani *and others*, 2017). One of the main applications is classifying single cells of a given tissue into specific cell subpopulations based on the genes they express. Single cell classification has been useful in identifying a new cell subtypes to refine existing classification (Villani *and others*, 2017). While such classification has been successfully accomplished through the development of numerous computational methods in recent years (Bacher and Kendziorowski, 2016), no statistical method has been developed for calculating the optimal number of cells that would allow identifying a new cell subtype.

There are mainly two challenges in calculating the number of cells required for identifying a new cell type. First, identifying a novel subpopulation from a known cell type population relies on unsupervised clustering methods such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) and UMAP (McInnes and Healy, 2018) while traditional sample size calculation is performed in the supervised hypothesis-testing framework (Hart *and others*, 2013).

Second, the statistical properties of scRNA-seq data are markedly different from those of bulk RNA-seq data. In particular, the number of genes with a zero read count is much higher in datasets for scRNA-seq than in bulk RNA-seq due to technical and/or biological reasons (Kharchenko *and others*, 2014). These dropout events need to be modeled properly to calculate an appropriate cell sample size.

Here we propose a statistical model of scRNA-seq data for calculating appropriate number of

cells for discovering a novel, rare cell type in a seemingly homogeneous population, which takes into account the characteristics of scRNA-seq data such as dropout events. Using this model, we estimated the population mean difference by projecting gene expressions onto a separating direction and used a hypothesis-testing framework to determine the separation of the two populations with relevant type I and type II error. We implemented our methods in an R package called *ncells* (<https://github.com/TheJacksonLaboratory/ncells>). In the Methods, we explain the details of our model, and in the Results, we applied *ncells* to the simulated scRNA-seq data and the scRNA-seq data on dendritic cells (DCs) and monocytes isolated from healthy blood donor (Villani *and others*, 2017) to show its efficacy in calculating the optimal number of cells in identifying a novel subpopulation. We also implemented a web-based pre-calculated version of *ncells* for selected input parameters that is available at <https://ncells.shinyapps.io/ncells>.

2. METHODS

ncells aim to estimate the number of cells required to distinguish a subpopulation from the population of cells that consist of two subpopulations Π_0 and Π_1 . The subpopulation Π_1 is separated from Π_0 using the marker genes which we define as upregulated genes in the subpopulation Π_1 . *ncells* use three key parameters that distinguish a subpopulation within a larger cell population: 1. the proportion of novel subpopulation Π_1 ; 2. the number of marker genes up-regulated in the subpopulation; 3. strength of marker gene expression (e.g. > 2-fold change increase in expression level). We first model gene expression without dropout event using a hierarchical normal distribution and then incorporate dropout events.

2.1 Modeling gene expression level in the absence of dropout event

We assume that log-transformed normalized read count values (e.g. $\log(TPM + 1)$) are normally distributed (Shalek *and others*, 2014; Law *and others*, 2014). Let Y_{ij} be the variance-standardized,

log-transformed count data of i th gene and j th cell. We then model expression level of each gene in a hierarchical manner.

For the non-marker gene i in cell j :

$$Y_{ij} \sim N(\mu_i, 1)$$

For the marker gene i :

$$Y_{ij} \sim \begin{cases} N(\mu_i + \log FC, 1) & \text{in cell } j \in \Pi_1 \\ N(\mu_i, 1) & \text{in cell } j \in \Pi_0 \end{cases}$$

where FC is the common fixed fold change of the marker genes. We assume each mean gene expression is distributed as normal with mean μ and variance σ^2 as

$$\mu_i \sim N(\mu, \sigma^2).$$

2.2 Modeling dropout events

In most scRNA-seq data, high proportions of zero read counts are observed due to biological or technical reasons. The rate of dropout events depends on the sequencing depth and thus varies greatly across different single-cell sequencing platforms. For example, scRNA-seq data obtained using Fluidigm C1 is known to have lower dropout rate on average than droplet based platforms such as inDrop or 10X Genomics (Li and Li, 2018). It is also presumed that the dropout event depends on the actual expression level (Marinov and others, 2014). Thus we model the probability of dropout event as being inversely proportional to actual expression values with a fixed rate α that takes into account platform-specific dropout rate:

$$\Pr(Y_{ij} \text{ is dropped out} | Y_{ij} = y_{ij}) = 1 - F(y_{ij})^\alpha$$

Here, F is the marginal cumulative distribution function of Y_{ij} obtained from the model in Section 2.1 as

$$F(y) = (1 - \pi_1 m_1) \Phi(y; \mu, \sigma^2 + 1) + \pi_1 m_1 \Phi(y; \mu + \log FC, \sigma^2 + 1)$$

where $\Phi(y; a, b)$ is the normal cumulative distribution with mean a and variance b . Therefore, dropout event depends both on the expression level and the parameter α that controls overall dropout rate. The overall dropout rate of the data is calculated as:

$$\Pr(\text{dropout}) = \frac{\alpha}{1 + \alpha}$$

For example, if the overall dropout rate is given as 70% a priori, α is set to 7/3.

2.3 Detecting a subpopulation

In our model without the dropout event, the difference between the two populations lies in the direction of the first principal component ($PC1$) (see Supplementary Information). With dropout event, $PC1$ still retains the population difference when the mean difference is large (See Supplementary Information for the relation of $PC1$ and the direction between the two population mean vector). Thus we can assess the separation of two populations by measuring the distance between the two sample means projected onto the sample $PC1$ and we define two clusters from the two populations as being separated when

$$|\gamma_1 - \gamma_0| > \delta \cdot s_{\text{pooled}}$$

where $\gamma_1, \gamma_0, s_{\text{pooled}}$ are two sample means and the pooled standard deviation of projected values onto $PC1$. Positive δ determines the degree of separation between two means γ_1 and γ_0 . Figure 1 illustrates an example of two subpopulations separated by $PC1$. We set the value of δ so that the type I error $\Pr(|\gamma_1 - \gamma_0| > \delta \cdot s_{\text{pooled}} | FC = 1)$ equals to the prespecified significance level, for example 0.05. For the more detailed discussion of $PC1$ in separating two populations, see the Supplementary Information.

2.4 Power calculation

Lastly, we estimate the probability of separation of the two populations through simulation, which serves as a power in conventional sample size calculation. The parameters required in the simulation include the mean μ and the standard deviation σ of the mean expression values, total number of cells sequenced (n), the proportion of cells in the subpopulation Π_1 (π_1), the total number of genes (p), the proportion of marker genes (m_1), the common fold change of the marker genes (FC), overall dropout rate (d) and significance level α . For the given parameters, we iterate simulations B times and estimate the power (probability of separation) by the frequency of separation of two sample populations out of B simulations:

$$\Pr(\text{separated}|\mu_0, \mu_1, n, \pi_1, d) \approx \frac{\#\text{separated}}{B} \quad (2.1)$$

where μ_0 and μ_1 are the mean expression vectors of Π_0 and Π_1 generated from parameters μ , σ , m_1 .

3. RESULTS

3.1 Simulation study

We performed simulations to assess statistical power to detect new subpopulation within a larger population of cells obtained from scRNA-seq experiment. We randomly selected the first $p \cdot m_1$ genes among p genes as the marker genes in the simulation. We simulate gene expression values using our model described in the Methods section with the following parameters.

- Mean μ of the mean expression values across genes: 1
- Standard deviation σ of the mean expression values across genes: 1
- Total number of cells (n): 100, 200, $\dots$, 1000
- Proportion of cells in the subpopulation Π_1 (π_1): 5%, 10%

- Total number of genes (p): 20000
- Proportion of marker genes (m_1): 0.1%, 0.2%, 0.4%, 0.8%, 1.6%
- Fold change in marker genes (FC): 1.5, 2, 4, 8
- Overall dropout rate (d): 60%, 70%, 80%, 90%, 95%

Figure 2 compares the effect of overall dropout rates when the percentage of marker gene is 0.8% with 2 fold change and the percentage of subpopulation = 10%, and $\mu = \sigma = 1$. It shows the general pattern of sample size and power: a larger sample size leads to a higher power (higher probability of finding the novel cells). Also as expected, it shows that a higher dropout rate requires more cells to achieve the same level of power. In the web-based *ncells* page (<http://ncells.shinyapps.io/ncells/>), it is shown that power increases by selecting higher marker gene proportions or by increasing fold change. From Figure 2, we can suggest that at least 300 cells would need to be sequenced to identify a subpopulation with 80% power, given 70% dropout rate. For additional results, see the web-based *ncells* page.

3.2 Real data analysis

We applied *ncells* to the dataset from Villani *and others* (2017) who performed single-cell RNA sequencing on dendritic cells (*DCs*) and monocytes isolated from a healthy blood donor. Previous studies classified human blood *DCs* into one plasmacytoid *DC* (*pDC*) and two *cDC* populations (*CD1C* and *CD141*). By using unsupervised analysis of single cell expression levels, they clustered 742 *DCs* derived from a single donor into 6 clusters, *DC1*, $\dots$, *DC6*, one of which is a newly discovered *DC* subpopulation (*DC5*). The *DC5* population is mixed with the previously classified *CD141* and *pDC* populations and we applied *ncells* to test if *DC5* can be distinguished within these two populations. They also clustered 339 monocytes into four subpopulations to identify two new monocyte clusters (*Mono3* and *Mono4*), which belong to the “intermediate”

monocyte population gated from $CD14^{++}CD16^{+}$ (hereafter *Mono intermediate* population). Using the parameters estimated from the data obtained from GEO (GSE94820), we estimated power and appropriate number of cells to detect *DC5* cluster in *CD141* and *pDC* population and *Mono3/Mono4* cluster in *Mono intermediate* population. The detailed information about the dataset and estimated parameters are in Table 1 and Supplementary Information.

We found that two out of the four clusters (*DC5* in *CD141*, *Mono4* in *Mono intermediate*) have sufficient powers ($> 95\%$) with total $n = 188$ and $n = 107$ cells, respectively while the other two clusters, *DC5* in *pDC* and *Mono3* in *Mono intermediate* didn't show sufficient power for total $n = 182$ and $n = 107$ cells. Relatively high variance and small proportion of subpopulation ($\sigma = 2.82, \pi_1 = 6.04$) may explain the low power of *DC5* cluster in *pDC* population. For *Mono3* cluster, the proportion of marker genes ($m_1 = 0.42$) may explain the low power. Detailed code and information is available in Supplementary information.

4. DISCUSSION

We developed a statistical model that incorporates key parameters of scRNA-seq data such as high dropout rate to estimate the number of cells required to identify a cell subpopulation within a seemingly homogeneous population. Within our model, we defined the separation of two populations with the relevant type I and II errors of the hypothesis-testing setup and calculated the optimal number of cells required to attain sufficient power. Our method does not evaluate any particular clustering method so it can be used as a general guide to design scRNA-seq experiment. We implemented the procedure in R package *ncells* and also web-based tool for some pre-selected parameters.

We modeled our dropout mechanism with cumulative marginal distribution of gene expression. The key idea was to take into account the protocol-dependent overall dropout rate and to have the dropout event depend only on the actual gene expression level. The overall dropout rate

is one crucial factor to determine the dropout effect in sample size calculation. However, there can be alternate approaches (such as logistic regression modeling in [Shalek and others, 2014](#)) and it would be useful to compare the effect of different dropout mechanisms in the sample size calculation.

We modeled gene expression levels using normal distribution with unit variance for gene expression distribution. Negative binomial model ([Robinson and others, 2010](#)) or zero-inflated negative binomial model ([Van De Wiel and others, 2013](#)) have been popular choice for modeling expression values of RNA-seq data. However, those models generally require more parameters to be specified than our model and since simulating high-dimensional gene expression data for many different conditions is computationally demanding task, it is important to reduce the number of parameters while keeping essential properties of scRNA-seq experiment. Further research is needed on more complex models but with computational feasibility.

For very high dropout rate (e.g. $> 97\%$), simulation based sample size calculation method becomes computationally demanding as it requires large number of samples. Recently dropseq based single cell RNA-seq technology (e.g. 10x Genomics) has been generating million cells in an experiment ([Zheng and others, 2017](#)). In this case, we can consider population level inference based on the asymptotic property of our model and estimate the optimal number of cells (See the discussion of Supplementary Information).

ncells is the first statistical method developed for calculating the number of cells required to detect a rare subpopulation in a homogeneous cell population for the given type I and II error. ***ncells*** will be an essential tool in designing single cell RNA sequencing experiment to study cell type classification.

REFERENCES

5. SOFTWARE

Software in the form of R code, together with a sample input data set and complete documentation is available on request from the corresponding author (kyungin.kim@jax.org).

6. SUPPLEMENTARY MATERIAL

ACKNOWLEDGMENTS

Conflict of Interest: None declared.

REFERENCES

- BACHER, RHONDA AND KENDZIORSKI, CHRISTINA. (2016). Design and computational analysis of single-cell rna-sequencing experiments. *Genome biology* **17**(1), 63.
- HART, STEVEN N, THERNEAU, TERRY M, ZHANG, YUJI, POLAND, GREGORY A AND KOCHER, JEAN-PIERRE. (2013). Calculating sample size estimates for rna sequencing data. *Journal of computational biology* **20**(12), 970–978.
- KHARCHENKO, PETER V, SILBERSTEIN, LEV AND SCADDEN, DAVID T. (2014). Bayesian approach to single-cell differential expression analysis. *Nature methods* **11**(7), 740–742.
- LAW, CHARITY W, CHEN, YUNSHUN, SHI, WEI AND SMYTH, GORDON K. (2014). Voom: precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology* **15**(2), R29.
- LI, WEI VIVIAN AND LI, JINGYI JESSICA. (2018). An accurate and robust imputation method scimpute for single-cell rna-seq data. *Nature communications* **9**(1), 997.

REFERENCES

11

- MAATEN, LAURENS VAN DER AND HINTON, GEOFFREY. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research* **9**(Nov), 2579–2605.
- MARINOV, GEORGI K, WILLIAMS, BRIAN A, McCUE, KEN, SCHROTH, GARY P, GERTZ, JASON, MYERS, RICHARD M AND WOLD, BARBARA J. (2014). From single-cell to cell-pool transcriptomes: stochasticity in gene expression and rna splicing. *Genome research* **24**(3), 496–510.
- McINNES, L. AND HEALY, J. (2018, February). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*.
- PATEL, ANOOP P, TIROSH, ITAY, TROMBETTA, JOHN J, SHALEK, ALEX K, GILLESPIE, SHAWN M, WAKIMOTO, HIROAKI, CAHILL, DANIEL P, NAHED, BRIAN V, CURRY, WILLIAM T, MARTUZA, ROBERT L, LOUIS, DAVID N, ROZENBLATT-ROSEN, ORIT, SUVÀ, MARIO L, REGEV, AVIV *and others*. (2014, 06). Single-cell rna-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science (New York, N.Y.)* **344**(6190), 1396–1401.
- RAMSKÖLD, DANIEL, LUO, SHUJUN, WANG, YU-CHIEH, LI, ROBIN, DENG, QIAOLIN, FARIDANI, OMID R, DANIELS, GREGORY A, KHREBTUKOVA, IRINA, LORING, JEANNE F, LAURENT, LOUISE C *and others*. (2012). Full-length mrna-seq from single cell levels of rna and individual circulating tumor cells. *Nature biotechnology* **30**(8), 777.
- ROBINSON, MARK D, MCCARTHY, DAVIS J AND SMYTH, GORDON K. (2010). edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**(1), 139–140.
- SHALEK, ALEX K, SATIJA, RAHUL, SHUGA, JOE, TROMBETTA, JOHN J, GENNERT, DAVE, LU, DIANA, CHEN, PEILIN, GERTNER, RONA S, GAUBLomme, JELLERT T, YOSEF, NIR *and others*. (2014). Single cell rna seq reveals dynamic paracrine control of cellular variation. *Nature* **510**(7505), 363.

- TANG, FUCHOU, BARBACIORU, CATALIN, WANG, YANGZHOU, NORDMAN, ELLEN, LEE, CLARENCE, XU, NANLAN, WANG, XIAOHUI, BODEAU, JOHN, TUCH, BRIAN B, SIDDIQUI, ASIM *and others*. (2009). mrna-seq whole-transcriptome analysis of a single cell. *Nature methods* **6**(5), 377–382.
- TASIC, BOSILJKA, MENON, VILAS, NGUYEN, THUC NGHI, KIM, TAE KYUNG, JARSKY, TIM, YAO, ZIZHEN, LEVI, BOAZ, GRAY, LUCAS T, SORENSEN, STACI A, DOLBEARE, TIM *and others*. (2016). Adult mouse cortical cell taxonomy by single cell transcriptomics. *Nature neuroscience* **19**(2), 335.
- VAN DE WIEL, MARK A, LEDAY, GWENAËL GR, PARDO, LUBA, RUE, HÅVARD, VAN DER VAART, AAD W AND VAN WIERINGEN, WESSEL N. (2013). Bayesian analysis of rna sequencing data by estimating multiple shrinkage priors. *Biostatistics* **14**(1), 113–128.
- VILLANI, ALEXANDRA-CHLOÉ, SATIJA, RAHUL, REYNOLDS, GARY, SARKIZOVA, SIRANUSH, SHEKHAR, KARTHIK, FLETCHER, JAMES, GRIESBECK, MORGANE, BUTLER, ANDREW, ZHENG, SHIWEI, LAZO, SUZAN *and others*. (2017). Single-cell rna-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science* **356**(6335), eaah4573.
- ZHENG, GRACE XY, TERRY, JESSICA M, BELGRADER, PHILLIP, RYVKIN, PAUL, BENT, ZACHARY W, WILSON, RYAN, ZIRALDO, SOLONGO B, WHEELER, TOBIAS D, MCDERMOTT, GEOFF P, ZHU, JUNJIE *and others*. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature communications* **8**, 14049.

REFERENCES

13

Fig. 1. . The first two principal component scores of gene expression values of 1000 cells. The first principal component ($PC1$) separates two populations of 900 (orange) and 100 (green) cells while the direction of the second principal component ($PC2$) does not show any separation. Common 2-fold change was applied to 100 marker genes.

Table 1. *Parameters used to calculate power for detection of the four clusters in Villani and others (2017). edgeR (Robinson and others, 2010) was used to detect differentially expressed genes with $FDR < 0.05$ and positive fold change in the subpopulation. Those differentially expressed genes were used as marker genes. Total number of genes (p) was 26593, estimated mean (μ) and standard deviation (σ) of mean gene expressions, the proportion of cells in the subpopulation (π_1), the proportion of marker genes (m_1), common fold change in the marker gene (FC), and overall dropout rate (d) were obtained from the dataset (See Supplementary Information).*

Population	σ	μ	n	$\pi_1(\%)$	$m_1(\%)$	FC	$d(\%)$
DC5 in <i>CD141</i>	1.03	1.90	188	9.04	1.59	2.87	79.0
DC5 in <i>pDC</i>	1.11	2.82	182	6.04	1.07	2.85	80.6
Mono3 in <i>Mono intermediate</i>	0.81	1.74	107	22.4	0.42	2.11	77.2
Mono4 in <i>Mono intermediate</i>	0.35	0.76	107	13.1	0.85	3.46	77.2

Fig. 2. Effect of dropout rate on the relationship between the power and cell sample size. Simulation parameters are: the proportion of marker genes $m_1 = 0.8\%$, the proportion of novel subpopulation $\pi_1 = 10\%$, 2-fold change in the marker gene expression, mean and variance $\mu = 1, \sigma = 1$ for the distribution of mean gene expressions, and total number of gene $p = 20000$.

REFERENCES

15

Fig. 3. Power and cell sample size for the parameters in Table 1 by *ncells*. The DC5 cluster in CD141 cells and Mono4 cluster in Mono intermediate cells were shown to have sufficiently high power (> 95% for $n = 188, n = 107$, respectively). DC5 cluster in pDC cells and Mono3 cluster in Mono intermediate cells were shown to have low power (< 45% for $n = 182$ and $\sim 30\%$ for $n = 107$, respectively).

Supplementary Information for “Calculating sample size for identifying cell subpopulation in single-cell RNA-seq experiments”, on *PC1*

Kyung In Kim, Ahrim Youn, Mohan Bolisetty,
A. Karolina Palucka, Joshy George

January 7, 2019

Without dropout event, the first principal component direction (*PC1*) is same as the direction that separates the two populations of interest. However, with dropout event when the difference between the two populations is small (e.g. small fold change, small number of marker genes etc), *PC1* may not represent the population separating direction. Instead it is likely that the *PC1* indicates the gene having the largest variance among all genes. Thus we investigate the condition for which *PC1* is the population separating direction. Also we provide relevance of *PC1* to identify subpopulations for varying parameters in our simulation studies.

PC1 in the absence of dropout event

As discussed in the Method section, we consider two populations Π_0 and Π_1 with proportions π_0 and π_1 , respectively ($\pi_0 + \pi_1 = 1$). Let M_0 and M_1 be the sets of non-marker and marker genes with proportions m_0, m_1 in total p genes ($m_0 + m_1 = 1$). The gene expression of i th gene in j th cell was defined as

$$y_{ij} \sim \begin{cases} N(\mu_i, 1) & \text{for } i \in M_0 \text{ or } i \in M_1, j \in \Pi_0 \\ N(\mu_i + \log FC, 1) & \text{for } i \in M_1, j \in \Pi_1 \end{cases} \quad (1)$$

where μ_i is the mean expression of i th gene and FC is the common fold change for the upregulated genes in M_1 . If we let $\mathbf{Y}$ be p -vector of gene

expressions and $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1$ be the mean expression vectors of Π_0, Π_1 , respectively, then the covariance matrix of $\mathbf{Y}$ is derived as:

$$\begin{aligned}\text{Var}(\mathbf{Y}) &= \text{Var}(E(\mathbf{Y}|Z)) + E(\text{Var}(\mathbf{Y}|Z)) \\ &= \text{Var}(Z\boldsymbol{\mu}_1 + (1-Z)\boldsymbol{\mu}_0) + E(Z\Sigma_1 + (1-Z)\Sigma_0) \\ &= \pi_0\pi_1\boldsymbol{\delta}\boldsymbol{\delta}' + \pi_0\Sigma_0 + \pi_1\Sigma_1\end{aligned}\quad (2)$$

where Z is the population indicator variable ($Z = 0, 1$) and $\boldsymbol{\delta}$ is $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ and Σ_k is the covariance matrix of population $k = 0, 1$. Both covariance matrices Σ_0, Σ_1 are identity matrices so $\text{Var}(\mathbf{Y}) = \pi_0\pi_1\boldsymbol{\delta}\boldsymbol{\delta}' + I$. Since for any eigenpair $(\lambda, \mathbf{e})$ of $\text{Var}(\mathbf{Y})$,

$$\pi_0\pi_1\boldsymbol{\delta}\boldsymbol{\delta}'\mathbf{e} + \mathbf{e} = \lambda\mathbf{e}$$

therefore $\pi_0\pi_1(\boldsymbol{\delta}'\mathbf{e})\boldsymbol{\delta} = (\lambda - 1)\mathbf{e}$. Thus, the largest eigenvalue is obtained when $\mathbf{e}$ is parallel to $\boldsymbol{\delta}$ and the eigenvalue is

$$1 + \pi_0\pi_1\|\boldsymbol{\delta}\|^2 = 1 + \pi_0\pi_1pm_1(\log FC)^2\quad (3)$$

Here the largest eigenvector $PC1$ is $\boldsymbol{\delta}/\|\boldsymbol{\delta}\|$. Therefore $PC1$ is the direction between the two population mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_0$.

PC1 in the presence of dropout event

Now consider the dropout mechanism as described in the Method section. Suppose $\tilde{y}$ is the gene expression of y in the presence of dropout. Then as defined in the Method section, we can write it as

$$\Pr(\tilde{y}|y) = \begin{cases} y & \text{with probability } F(y)^\alpha \\ 0 & \text{with probability } 1 - F(y)^\alpha \end{cases}$$

thus, the marginal expectation of $\tilde{y}$ is

$$E(\tilde{y}) = \int E(\tilde{y}|y)f(y)dy = \int yF(y)^\alpha f(y)dy$$

where $f(y)$ is the density of y before dropout event. Since we assume the dropout event is only dependent on the expression value y , we can still use the same covariance matrix equation in (2) but with different mean vectors, $\tilde{\boldsymbol{\mu}}_0, \tilde{\boldsymbol{\mu}}_1$ and the covariance matrices, $\tilde{\Sigma}_0$ and $\tilde{\Sigma}_1$. To compute those values,

we first need to compute each entry of those vectors. Let $\tilde{\mu}_i^k$ and $\tilde{\sigma}_i^k$ be the mean and the variance of i th gene in population k then we have

$$\tilde{\mu}_i^k = \int y F(y)^\alpha f_{ik}(y) dy, \quad (4)$$

$$\tilde{\sigma}_i^k = \int y^2 F(y)^\alpha f_{ik}(y) dy - (\tilde{\mu}_i^k)^2 \quad (5)$$

where f_{ik} is the density of the i th gene in Π_k as in (1). F and α are same as described in the Method section. From (2), we have the covariance matrix in the presence of dropout as

$$\text{Var}(\tilde{\mathbf{Y}}) = \pi_0 \pi_1 \tilde{\boldsymbol{\delta}} \tilde{\boldsymbol{\delta}}' + \pi_0 \tilde{\Sigma}_0 + \pi_1 \tilde{\Sigma}_1 \quad (6)$$

where the i th entry of $\tilde{\boldsymbol{\delta}}$ is $\tilde{\mu}_i^1 - \tilde{\mu}_i^0$ and the i th diagonal entry of diagonal matrix $\tilde{\Sigma}_k$ is $\tilde{\sigma}_i^k$ for $k = 0, 1$. Note that $\tilde{\sigma}_i^k$ depends on μ_i so $\tilde{\Sigma}_k$ is not necessarily identity matrix.

The main concern is when the largest eigenvector ($PC1$) of (6) is able to separate the two populations of interest. In other words, we need to investigate when $PC1$ of (6) is not orthogonal to $\tilde{\boldsymbol{\delta}}$, the population separating direction.

Now consider a decomposition of $\tilde{\mathbf{Y}}$ into non-marker and marker genes, say $\tilde{\mathbf{Y}}_{M_0}$ and $\tilde{\mathbf{Y}}_{M_1}$. Then the corresponding mean difference vector $\tilde{\boldsymbol{\delta}}_{M_0}$ of non-marker set M_0 is zero vector and

$$\begin{aligned} \text{Var}(\tilde{\mathbf{Y}}) &= \pi_0 \pi_1 \begin{bmatrix} \tilde{\boldsymbol{\delta}}_{M_1} \tilde{\boldsymbol{\delta}}_{M_1}' & 0 \\ 0 & 0 \end{bmatrix} + \pi_0 \begin{bmatrix} \tilde{\Sigma}_{0,M_1} & 0 \\ 0 & \tilde{\Sigma}_{0,M_0} \end{bmatrix} + \pi_1 \begin{bmatrix} \tilde{\Sigma}_{1,M_1} & 0 \\ 0 & \tilde{\Sigma}_{0,M_0} \end{bmatrix} \\ &\equiv \begin{bmatrix} \text{Var}(\tilde{\mathbf{Y}}_{M_1}) & 0 \\ 0 & \text{Var}(\tilde{\mathbf{Y}}_{M_0}) \end{bmatrix} \end{aligned}$$

Since $\text{Var}(\tilde{\mathbf{Y}})$ has a block diagonal structure, the eigenvectors of $\text{Var}(\tilde{\mathbf{Y}})$ can be obtained separately from $\text{Var}(\tilde{\mathbf{Y}}_{M_1})$ and $\text{Var}(\tilde{\mathbf{Y}}_{M_0})$ and the two sets of eigenvectors are orthogonal to each other. Therefore, if we let λ_{M_0} and λ_{M_1} be the largest eigenvalues of $\text{Var}(\tilde{\mathbf{Y}}_{M_0})$ and $\text{Var}(\tilde{\mathbf{Y}}_{M_1})$, then a necessary condition for $PC1$ separating the two populations (or not orthogonal to $\tilde{\boldsymbol{\delta}}$) is $\lambda_{M_1} > \lambda_{M_0}$.

We claim that $\lambda_{M_1} > \lambda_{M_0}$ is also a sufficient condition for $PC1$ separating the two populations.

Proof. For simplicity, let $\text{Var}(\tilde{\mathbf{Y}}_{M_1}) = \mathbf{a}\mathbf{a}' + D$ where $\mathbf{a} = \sqrt{\pi_0\pi_1}\tilde{\boldsymbol{\delta}}_{M_1}$ and $D = \pi_0\tilde{\Sigma}_{0,M_1} + \pi_1\tilde{\Sigma}_{1,M_1}$ and let $\mathbf{e}$ is the largest eigenvector. Suppose that $\mathbf{e}$ and $\mathbf{a}$ are orthogonal. Then since $\mathbf{e}$ is an eigenvector,

$$\text{Var}(\tilde{\mathbf{Y}}_{M_1})\mathbf{e} = \mathbf{a}\mathbf{a}'\mathbf{e} + D\mathbf{e} = \lambda_{M_1}\mathbf{e} \implies D\mathbf{e} = \lambda_{M_1}\mathbf{e}$$

implies $\mathbf{e}$ is an eigenvector of D . Since D is a diagonal matrix, all eigenvectors are the Euclidean coordinate vectors of D , which in turn $\mathbf{e}$ is one of them, say i th coordinate vector. Then the orthogonality assumption implies i th entry of $\mathbf{a}$, $\mathbf{a}'\mathbf{e}$ is zero. However it contradicts to the fact that $\mathbf{a}'\mathbf{e} = \sqrt{\pi_0\pi_1}(\tilde{\mu}_i^1 - \tilde{\mu}_i^0)$ is not zero by construction in (4). $\square$

We then investigate the factors that determine λ_{M_0} and λ_{M_1} . Since λ_{M_0} is the maximum entry of diagonals $\tilde{\sigma}_i^k$ s for $i \in M_0$ (see (5)), we can see that λ_{M_0} is largely affected by the variation of the mean values of individual non-marker genes. For λ_{M_1} , the characteristic polynomial of $\text{Var}(\tilde{\mathbf{Y}}_{M_1})$ is

$$\begin{aligned} \det(\lambda I - \text{Var}(\tilde{\mathbf{Y}}_{M_1})) &= \det(\lambda I - \mathbf{a}\mathbf{a}' - D) \\ &= \det(\lambda I - D)(1 - \mathbf{a}'(\lambda I - D)^{-1}\mathbf{a}) \\ &= \left(\prod_{i \in M_1} (\lambda - d_i) \right) \left(1 - \sum_{i \in M_1} \frac{a_i^2}{\lambda - d_i} \right) \end{aligned}$$

where $\mathbf{a} = \sqrt{\pi_0\pi_1}\tilde{\boldsymbol{\delta}}_{M_1} = (a_i)_{i \in M_1}$ and $D = \pi_0\tilde{\Sigma}_{0,M_1} + \pi_1\tilde{\Sigma}_{1,M_1} = \text{diag}(d_i)_{i \in M_1}$. As far as we know, there is no analytic solution for the largest root of the characteristic polynomial. So as a special case we assume $d_i = d$ for all i . Then for an eigenvalue λ it satisfies

$$\left(\prod_{i \in M_1} (\lambda - d) \right) \left(1 - \sum_{i \in M_1} \frac{a_i^2}{\lambda - d} \right) = 0$$

and it implies

$$\lambda_{M_1} = d + \sum_{i \in M_1} a_i^2 = d + \pi_0\pi_1 \sum_{i \in M_1} (\tilde{\mu}_i^1 - \tilde{\mu}_i^0)^2 = d + \pi_0\pi_1 \|\tilde{\boldsymbol{\delta}}_{M_1}\|^2$$

so the magnitude of λ_{M_1} is largely determined by $\tilde{\boldsymbol{\delta}}_{M_1}$ and π_1 (Compare this with (3)).

Rank of λ_{M_1}

When $\lambda_{M_1} < \lambda_{M_0}$, the $PC1$ is not informative in separating the two populations of interest as it is orthogonal to $\tilde{\delta}$. In this case, to detect the population separating direction, we need more than one PC s and we might need to develop hypothesis testing setup based on the selected multiple PC s. It is obvious that the minimal number of PC s required in this scenario is the rank of λ_{M_1} among the eigenvalues of $\text{Var}(\tilde{\mathbf{Y}})$.

We calculated the rank of λ_{M_1} for our simulation study in Supplementary Table 1 (For simulation results, see <https://ncells.shinyapps.io/ncells>) where $\mu = 1$ and $\sigma = 1$ of mean gene expression were used. In most conditions, λ_{M_1} is larger than λ_{M_0} , however when FC or marker gene fraction m_1 is small, the ranks of λ_{M_1} become larger than λ_{M_0} . In those cases, $PC1$ has no information for separating two populations so one can avoid further simulations to calculate powers for the conditions. Function `ncomps` in the R package *ncells* was used to calculate the rank of λ_{M_1} .

Supplementary Table 1: Rank of λ_{M_1} among the eigenvalues of $\text{Var}(\tilde{\mathbf{Y}})$. μ_0 was generated from $N(0, 1)$ and all entries of μ_1 are same as μ_0 except the first $20000 \cdot m_1$ entries with difference $\log FC$. m_1 is the proportion of marker genes. π_1 is the proportion of minor subpopulation. FC is the fold change upregulated in the minor subpopulation. d is the overall dropout rate.

			$d = 0.6$	$d = 0.7$	$d = 0.8$	$d = 0.9$	$d = 0.95$
$m_1 = 0.001$	$\pi_1 = 0.05$	$FC = 1.5$	1	1	514	670	679
		$FC = 2$	1	1	1	512	580
		$FC = 4$	1	1	1	1	46
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.1$	$FC = 1.5$	1	1	199	577	610
		$FC = 2$	1	1	1	238	400
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.05$	$FC = 1.5$	1	1	486	668	679
		$FC = 2$	1	1	1	426	564
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
$m_1 = 0.002$	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	565	607
		$FC = 2$	1	1	1	1	350
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.1$	$FC = 1.5$	1	1	1	80	256
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	238	278
		$FC = 2$	1	1	1	1	213
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
$m_1 = 0.004$	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	36	256
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.1$	$FC = 1.5$	1	1	1	1	161
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
$m_1 = 0.008$	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.1$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
$m_1 = 0.016$	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.1$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1
	$\pi_1 = 0.05$	$FC = 1.5$	1	1	1	1	1
		$FC = 2$	1	1	1	1	1
		$FC = 4$	1	1	1	1	1
		$FC = 8$	1	1	1	1	1

Supplementary Information for “Calculating sample size for identifying cell subpopulation in single-cell RNA-seq experiments”, Real data analysis

Kyung In Kim, Ahrim Youn, Mohan Bolisetty, A. Karolina Palucka, Joshy George

2019-01-04

```
library(Biobase)
library(readxl)
library(knitr)
library(kableExtra)
library(ggplot2)
library(reshape2)
library(limma)
library(tibble)
library(ggfortify)
library(tidyverse)
library(edgeR)
library(openxlsx)
```

```
path <- list()
path$data <- "../Data/GSE94820"
```

- We downloaded single cell dataset from Gene Expression Omnibus (GSE94820). Three known cell populations (two dendritic cell populations, CD141 and pDC and one monocyte cell population named as Mono_intermediate) of the dataset were considered for our study. Based on unsupervised clustering, Villani et al. (2017) found 10 clusters from the three populations and three clusters, DC5, Mono3, and Mono4 were selected to investigate as new subpopulations of the three known populations. We first assume those three clusters are new subpopulations and estimate the parameters for simulation using differential expression analysis (edgeR).

```
gse94820 <- paste(path$data, "GSE94820_cpm.RData", sep = "/")
load(gse94820)
```

```
x <- with(pData(eset_cpm), rbind(table(celltype, RNAseq), table(subtype,
  RNAseq)))
rownames(x)[7] <- "Mono_intermediate"
kable(x[c(1, 5, 7), ])
```

	DC1	DC2	DC3	DC4	DC5	DC6	Mono1	Mono2	Mono3	Mono4
CD141	165	4	0	0	17	0	1	0	0	1
pDC	0	0	0	0	11	171	0	0	0	0
Mono_intermediate	0	1	0	1	0	2	53	12	24	14

```
diffanal <- function(eset, alpha) {
  y <- DGEList(counts = exprs(eset), group = pData(eset)$population)
  keep <- rowSums(cpm(y) > 1) >= 2
  y <- y[keep, keep.lib.sizes = FALSE]
  y <- calcNormFactors(y)
  y <- estimateDisp(y)
```

```

tab <- exactTest(y)$table
tab$FDR <- with(tab, p.adjust(PValue, "BH"))
sigtab <- with(tab, tab[FDR < alpha & logFC > 0, ])

p <- nrow(eset)
p1 <- nrow(sigtab)/p * 100
dropout <- (1 - sum(exprs(eset) > 0)/prod(dim(eset))) * 100
return(list(p = p, p1 = p1, dropout = dropout, tab = sigtab))
}

estimateFC <- function(eset, siggenes) {
  emat <- log2(exprs(eset[siggenes, ]) + 1)
  x0 <- with(pData(eset), emat[, population == 0])
  x1 <- with(pData(eset), emat[, population == 1])
  n0 <- ncol(x0)
  n1 <- ncol(x1)
  s0 <- apply(x0, 1, sd)
  s1 <- apply(x1, 1, sd)
  sp <- sqrt(((n0 - 1) * s0 + (n1 - 1) * s1)/(n0 + n1 - 2))

  y0 <- apply(x0, 2, function(x) x/sp)
  y1 <- apply(x1, 2, function(x) x/sp)
  m0 <- rowMeans(y0)
  m1 <- rowMeans(y1)

  return(list(fc = 2^mean(m1 - m0), mu = mean(m0), sigma = sd(m0)))
}

npop <- 4
param <- vector("list", npop)
names(param) <- c("DC5_CD141", "DC5_pDC", "Mono3_Mono_intermediate",
  "Mono4_Mono_intermediate")

```

DC5 subpopulation in CD141 cells

- Total number of cells in CD141: 141
- Number of cells in DC5: 17

```

eset_sel <- with(pData(eset_cpm), eset_cpm[, celltype == "CD141"])
pData(eset_sel)$population <- with(pData(eset_sel), ifelse(RNAseq ==
  "DC5", 1, 0))

ans <- diffanal(eset_sel, 0.05)
ans$n0 <- with(pData(eset_sel), sum(population == 0))
ans$n1 <- with(pData(eset_sel), sum(population == 1))
est <- estimateFC(eset_sel, rownames(ans$tab))
ans$fc <- est$fc
ans$mu <- est$mu
ans$sigma <- est$sigma

param[[1]] <- ans

```

DC5 subpopulation in pDC cells

- Total number of cells in pDC: 11
- Number of cells in DC5: 182

```
eset_sel <- with(pData(eset_cpm), eset_cpm[, celltype == "pDC"])
pData(eset_sel)$population <- with(pData(eset_sel), ifelse(RNAseq ==
  "DC5", 1, 0))

ans <- diffanal(eset_sel, 0.05)
ans$n0 <- with(pData(eset_sel), sum(population == 0))
ans$n1 <- with(pData(eset_sel), sum(population == 1))
est <- estimateFC(eset_sel, rownames(ans$tab))
ans$fc <- est$fc
ans$mu <- est$mu
ans$sigma <- est$sigma

param[[2]] <- ans
```

Mono3 subpopulation in Mono_intermediate cells

- Total number of cells in Mono_intermediate: 107
- Number of cells in Mono3: 24

```
eset_sel <- with(pData(eset_cpm), eset_cpm[, celltype == "Mono" &
  subtype == "intermediate"])
pData(eset_sel)$population <- with(pData(eset_sel), ifelse(RNAseq ==
  "Mono3", 1, 0))

ans <- diffanal(eset_sel, 0.05)
ans$n0 <- with(pData(eset_sel), sum(population == 0))
ans$n1 <- with(pData(eset_sel), sum(population == 1))
est <- estimateFC(eset_sel, rownames(ans$tab))
ans$fc <- est$fc
ans$mu <- est$mu
ans$sigma <- est$sigma

param[[3]] <- ans
```

Mono4 subpopulation in Mono_intermediate cells

- Total number of cells in Mono_intermediate: 107
- Number of cells in Mono4: 14

```
eset_sel <- with(pData(eset_cpm), eset_cpm[, celltype == "Mono" &
  subtype == "intermediate"])
pData(eset_sel)$population <- with(pData(eset_sel), ifelse(RNAseq ==
  "Mono4", 1, 0))

ans <- diffanal(eset_sel, 0.05)
ans$n0 <- with(pData(eset_sel), sum(population == 0))
ans$n1 <- with(pData(eset_sel), sum(population == 1))
est <- estimateFC(eset_sel, rownames(ans$tab))
```



```
ans$fc <- est$fc
ans$mu <- est$mu
ans$sigma <- est$sigma

param[[4]] <- ans
```

Summary

- Below is the all the parameters estimated from the single cell dataset.

```
info <- data_frame(id = names(param))
info$p <- sapply(param, function(x) x$p)
info$`marker (%)` <- round(sapply(param, function(x) x$p1), 2)
info$`dropout (%)` <- round(sapply(param, function(x) x$dropout),
  2)
info$mu <- round(sapply(param, function(x) x$mu), 2)
info$sigma <- round(sapply(param, function(x) x$sigma), 2)
info$n <- sapply(param, function(x) x$n0 + x$n1)
info$`subpopulation (%)` <- round(sapply(param, function(x) x$n1/(x$n0 +
  x$n1) * 100), 2)
info$fold_change <- round(sapply(param, function(x) x$fc), 2)

kable(info, "latex", booktabs = TRUE) %>% kable_styling(latex_options = c("striped",
  "scale_down"))
```

id	p	marker (%)	dropout (%)	mu	sigma	n	subpopulation (%)	fold_change
DC5_CD141	26593	1.59	78.96	1.03	1.90	188	9.04	2.87
DC5_pDC	26593	1.07	80.57	1.11	2.82	182	6.04	2.85
Mono3_Mono_intermediate	26593	0.42	77.20	0.81	1.74	107	22.43	2.11
Mono4_Mono_intermediate	26593	0.85	77.20	0.35	0.76	107	13.08	3.46

Sample size plots for four populations

- `res` object in the code below is the output of `ncells` function in R package `ncells` version 0.3.0, which was subsequently stored in `RData` file and is called to calculate the power (Current version 0.4.0 directly calculates the power given cut-off values and other conditions). `null-population.RData` has the output `res` of the function `ncells` with parameter $FC = 1$ and `nonnull-population.RData` has `res` with parameter $FC > 1$.

```
colnames(info) <- c("id", "p", "m1", "dp", "mu", "sigma", "n",
  "pi1", "fc")

pops <- info$id

spooled <- function(s0, s1, n0, n1) return(sqrt((s0^2 * (n0 -
  1) + s1^2 * (n1 - 1))/(n0 + n1 - 2)))
nt <- seq(50, 500, by = 50)
dfactor <- matrix(, length(pops), length(nt))
rownames(dfactor) <- pops
colnames(dfactor) <- sprintf("n=%s", nt)
```

```

alpha <- 0.05
for (k in 1:length(pops)) {
  load(sprintf("%s/null-%s.RData", path$data, pops[k]))
  for (m in 1:length(nt)) {
    x <- res[, , m]
    r <- abs(x[, "m0"] - x[, "m1"])/spooled(x[, "s0"], x[,
      "s1"], (1 - info$pi1[k]/100) * nt[m], info$pi1[k]/100 *
      nt[m])
    w <- seq(0, max(r), by = 0.01)
    fw <- sapply(w, function(x) mean(r > x))
    g <- splinefun(x = w, y = fw, method = "monoH.FC")
    h <- function(x) g(x) - alpha
    dfactor[k, m] <- uniroot(h, interval = c(0, max(r)))$root
  }
}

rate <- matrix(, length(pops), length(nt))
rownames(rate) <- pops
colnames(rate) <- sprintf("n=%s", nt)
for (k in 1:length(pops)) {
  load(sprintf("%s/nonnull-%s.RData", path$data, pops[k]))
  for (m in 1:length(nt)) {
    x <- res[, , m]
    rate[k, m] <- mean(abs(x[, "m0"] - x[, "m1"]) > dfactor[k,
      m] * spooled(x[, "s0"], x[, "s1"], (1 - info$pi1[k]/100) *
      nt[m], info$pi1[k]/100 * nt[m]))
  }
}

dfrate <- cbind(`Number of Cells` = factor(sprintf("n=%s", seq(50,
  500, by = 50))), levels = sprintf("n=%s", seq(50, 500, by = 50))),
  as_tibble(t(rate)))
dfrate_melt <- melt(dfrate)
colnames(dfrate_melt)[2:3] <- c("Population", "Power")

dfrate_melt <- subset(dfrate_melt, !grepl("_no_", Population))

p <- ggplot(dfrate_melt, aes(x = `Number of Cells`, y = Power,
  colour = Population, group = Population))
p <- p + geom_line(aes(linetype = Population)) + geom_point(aes(shape = Population))
p <- p + theme(axis.text.x = element_text(angle = 45, hjust = 1))
p <- p + scale_y_continuous(breaks = seq(0, 1, by = 0.1), limit = c(0,
  1))
p

```