CellexalVR: A virtual reality platform for the visualisation and analysis of single-cell gene expression data

Oscar Legetth[1,2], Johan Rodhe[1,2], Joel Pålsson[3], Mattias Wallergård[4], Stefan Lang[1,2], and Shamit Soneji[1,2,*]

[1]Division of Molecular Hematology, BMC, Lund University, Lund Sweden

[2]Lund Stem Cell Center, Lund University, Lund, Sweden

[3]LTH, Lund University, Lund, Sweden

[4]Department of Design Sciences, Lund university

[*]Corresponding author

Single-cell RNAseq is a powerful tool for the dissection of cell populations. Multiple dimension reduction (DR) tools are available to project cells on to 3-dimensional space that allow one to visualise the heterogeneity within the assayed population, often forming complex cellular maps. Thus far, visualisation methods for 3D embedded maps are poor, and the lack of intuitive point/cell selection often hinders a rapid exploration of finer details contained in the data. Moreover, directly comparing the output from several DR methods is not possible. Here we present CellexalVR (www.cellexalvr.med.lu.se), a feature-rich, fully interactive virtual reality environment for the visualisation and analysis of single-cell RNAseq experiments that allows researchers to intuitively and collaboratively gain an understanding of their data.

Single-cell RNAseq (scRNAseq) is a routinely used method to explore the heterogeneity of

cell populations. As a consequence the quantity of scRNAseq data is growing rapidly with the number of cells assayed per experiment projected to increase similarly [9] as technology improves. Projects such as the Human Cell Atlas [7] will single-cell profile a massive number of cells that will be of general interest to the wider scientific community, and methods are needed for researchers of varied computational ability to access and explore it.

The analysis of single-cell RNAseq data is often performed using scripting, primarily using packages for the R/Python languages such as monocle [10], Seurat [2] and SCANPY [12] among others. A common step after pre-processing is a dimension-reduction (DR) where the cells are positioned in 2-3 dimensional space to visualise heterogeneity within the assayed populations. A number of methods are available to do this, tSNE [11], diffusion maps [3] and UMAP [1] among others, all of which result in different projections given their differing underlying methods. Typically this means several methods are used during the course of an analysis and compared. Cells are often embedded into 3-dimensions because it allows greater visual power when resolving the spatial arrangement of the cells and the clusters they form, and indeed, when single-cell experiments are reported an increasingly common practice is to provide a webtool where 3D DR plots can be coloured by gene/cell information and explored in a rudimentary fashion. The visualisation of 3D embedded data is thus restricted to 2D computer displays (using OpenGL for example), and this has several shortcomings. These plots have limited viewing angles since they have only one point of rotation at the centre which makes it difficult to view specific sub-populations on the periphery, but importantly, only one projection can be loaded in the same window meaning direct comparisons between several projection methods cannot be made easily. The main drawback is the plot is still 2D when viewed on a conventional desktop display as there is no sense of depth, thus complex projections and the finer structures within are harder to comprehend. Furthermore these plots are not interactive, so selecting cells for further analyses is not possible.

Here we present CellexalVR, a virtual reality (VR) platform that overcomes these issues. By placing all dimension-reduced (DR) representations of the data in VR we have created an immersive environment to explore and analyse scRNAseq experiments. In VR 3D DR plots

have depth and can be interacted with and manipulated intuitively, for example, grasping and moving them to gain any view required as if they were a physical object being held in the hand. Multiple DR plots derived from different methods or the same methods but using different parameters can be loaded in a single session and cross-compared with ease. For example, cells of interest in a tSNE plot can be selected and traced to their counterparts in a diffusion map allowing the user to directly visualise the differences between the two reduction methods. Another use of this feature would be to determine the effect of different pre-processing steps on the outcome when the same DR method is applied downstream.

CellexalVR comprises of two components. The first is the VR interface which has been implemented using the Unity game engine (https://unity3d.com/), and an R package (cellexalvrR) (https://github.com/sonejilab/cellexalvrR) that performs two functions. The first is to undertake back-end calculations during a CellexalVR session (calculating differentially expressed genes and correlation networks among others), and the second is to provide simple functions that allow the user to export their scRNAseq data from an R session to a set of input files that CellexalVR can read. Figure 1 shows a simplified workflow where scRNAseq data is pre-processed using user-preferred methods, and the required files are created using the functions provided by the cellexalvrR package. Figure S1 shows a mixed reality shot taking during a session that shows the data in relation to a user. The reason for compartmentalising CellexalVR is so bioinformaticians can alter the R package to modify/add computational methods without needing knowledge of C# (the language used by Unity). At a minimum CellexalVR should be provided with the gene expression data (a subset of variable genes are recommended), and one set of DR coordinates. CellexalVR will also import cell surface marker intensities captured during index sorting/CITEseq, and categorical metadata for cells and genes. Detailed documentation and instructional videos are provided on the project website.

Figure 2 shows a selection of features available in CellexalVR using data from mouse hematopoietic stem and progenitor cells [5]. DR plot(s) are automatically loaded when a new CellexalVR session is initiated, and when more than one is present they are all loaded

simultaneously (Fig 2a and Movie 1). These can be coloured according to the expression of a selected gene (here, Gata1) (Fig 2b and Movie 2) using a keyboard in the virtual environment. A prominent feature of CellexalVR is the ability to freehand capture cells of interest by passing them through a selection tool that extends from the action controller when activated (Fig 2c, Movie 3). As cells pass through the tool they are coloured as they are selected, and their counterpart in other DR plots in the session are coloured simultaneously, thus giving instant feedback as to where these cells reside in other projections. A new group is initiated by a left/right-click on the controller's touchpad that changes the tool to a new colour. Once the desired groups have been captured and confirmed one can produce a heatmap of the top *N* differentially expressed genes that are calculated in R and then rendered in the VR environment (Fig 2d, Movie 4). The heatmap can be moved/resized/rearranged, but importantly, the order of the cells in the heatmap is defined by the order in which they are passed through the selection tool, and this is particularly useful when selecting through pseudotime projections as this cell-order information is preserved. Clicking on a gene name in the heatmap will recolour the DR plots by the expression of the gene selected. A second option is to generate transcription factor correlation networks from each of the defined groups, but as they are in the same virtual space they can be compared directly to one another to visualise which TF-TF pairs are in common between each of the different networks (Figure S2 and Movie 6). A third and powerful option is to trace the selected cells to their counterparts in other DR plots. Fig 2e and Movie 7 show a group of cells that have been selected in the leftmost DDRTree projection and then traced to their counterparts in the tSNE plot on the right. These seemingly similar cells actually split into two further groups when both projections are considered together, and these two groups can be captured by passing the selection tool over cubes placed on the connecting lines. Meta-data for cells can also be overlaid with ease using the attributes menu on the controller (Figure S3). For very dense maps with a large number of cells this can be problematic. Figure 2f shows 116,000 cells from mouse gastrulation [6] (left), and coloured are two mesoderm types. The space occupied by these groups is difficult to see when obscured by uncoloured cells, so to overcome this they can be rendered in a separate plot within a skeleton of the total map making it much clearer how these types relate to each other (Movie 8).

Cell surface marker information (whether it be measured via index sorting or CITEseq [8]) can also be imported and used in CellexalVR. Cells can be coloured by the expression of a selected marker (Figure S4) and markers can also be plotted against each other from which populations can be "gated" using the selection tool. These cells can be traced to their counterparts in the DR plots, but also selected for further analysis using an alternative set of surface markers (Figure S5).

CellexalVR is built on video game technology, we have taken advantage of this and introduced multi-user mode which utilises the Photon Unity Network (PUN) where users can meet in CellexalVR and analyse a dataset together regardless of geographical location. Voice is transmitted by a third-party application, and we recommend Discord (https://discordapp.com/) which is a low-latency voice over IP application used by gamers. Figure 3 shows two users in a CellexalVR session where both have complete access to the data and the tools to analyse it. Users can also join a session in "ghost mode", i.e they are present in the session but do not have a visible physical presence (head/hands) to reduce clutter in the work space. Ghost participants do not have the ability to interact with the data but can still communicate by voice. There is no limit on the number of participants in a session. Multi-user mode represents a massive advance in how scientists can work collaboratively on scientific data without restrictions that plague conventional desktop conferencing software/screen sharing systems. All of these functions are triggered using one touch operations on the controller mounted menu system (Fig 2a, bottom left and FigS3). Figures generated during a session can be exported as png images, and cell selections can be exported as text files. CellexalVR will take data from any scRNAseq method, and the current limit for the number of cells that can be displayed in a single field-of-view is approximately 250,000 after engineering a new collision detection method utilising octrees [4] and axis aligned bounding boxes (AABB).
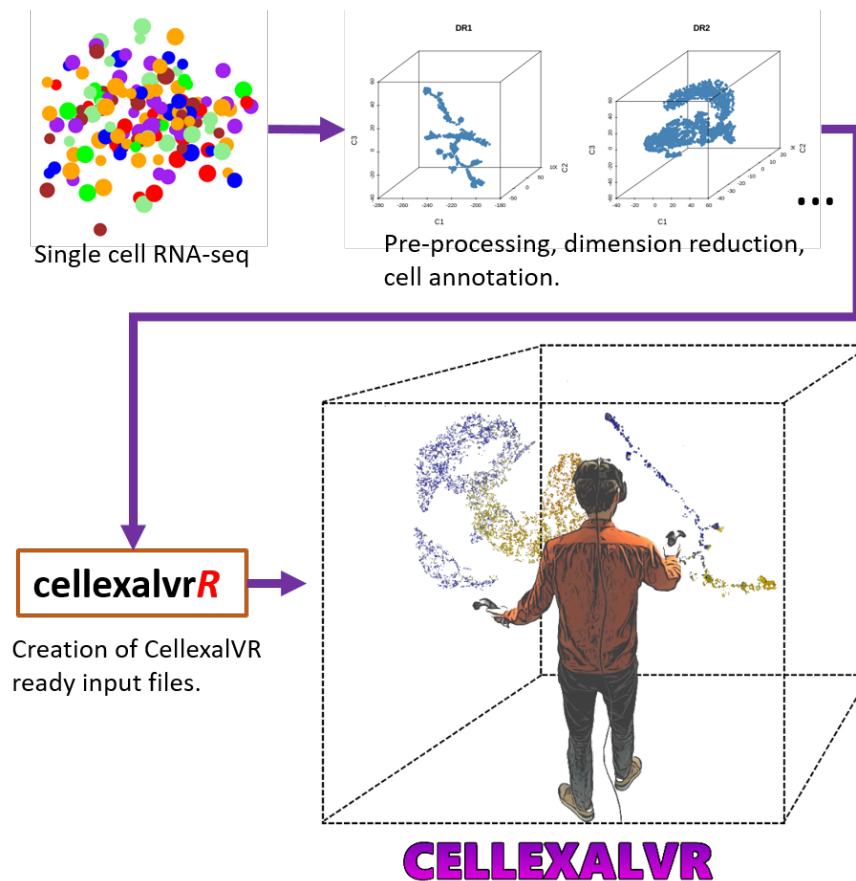
Figure 1: Schematic overview of CellexalVR. Single-cell data is processed using user-preferred pipelines/methods after which the appropriate matrices are formatted by the R package cellexalvrR to produce the input files and database required by CellexalVR. Bottom-right shows the data in relation to the user whom is wearing the VR headset and is holding a controller in each hand.

In order to expedite the learning process we have extensive documentation and tutorial videos on the project website. CellexalVR has in-session help, and first time users can take an introductory hands-on tutorial "level" designed to familiarise users with the controls. Our testers became proficient in approximately 30 minutes regardless of age, video game or VR experience. With the number of scRNAseq datasets predicted to increase rapidly, alternative intuitive methods are required to navigate it and we have shown that virtual reality is an extremely attractive solution for the visualisation of big-data.

Availability and system requirements.

CellexalVR is available from the project website at https://cellexalvr.med.lu.se/download and the R backend can be installed directly from GitHub https://github.com/sonejilab/cellexalvrR. Users will need a gaming-class computer with a high-end graphics card (for

example an NVIDIA GTX1080) running Windows 10, and an HTC Vive VR kit comprising of the headset, controllers and base stations. We recommend the Vive Pro that has a higher resolution that makes reading text easier.

Author contributions.

O.L, J.R, and J.P developed CellexalVR and the project website. S.L and S.S developed cellexalvrR. M.W provided technical assistance. S.S conceived the study and wrote the paper.
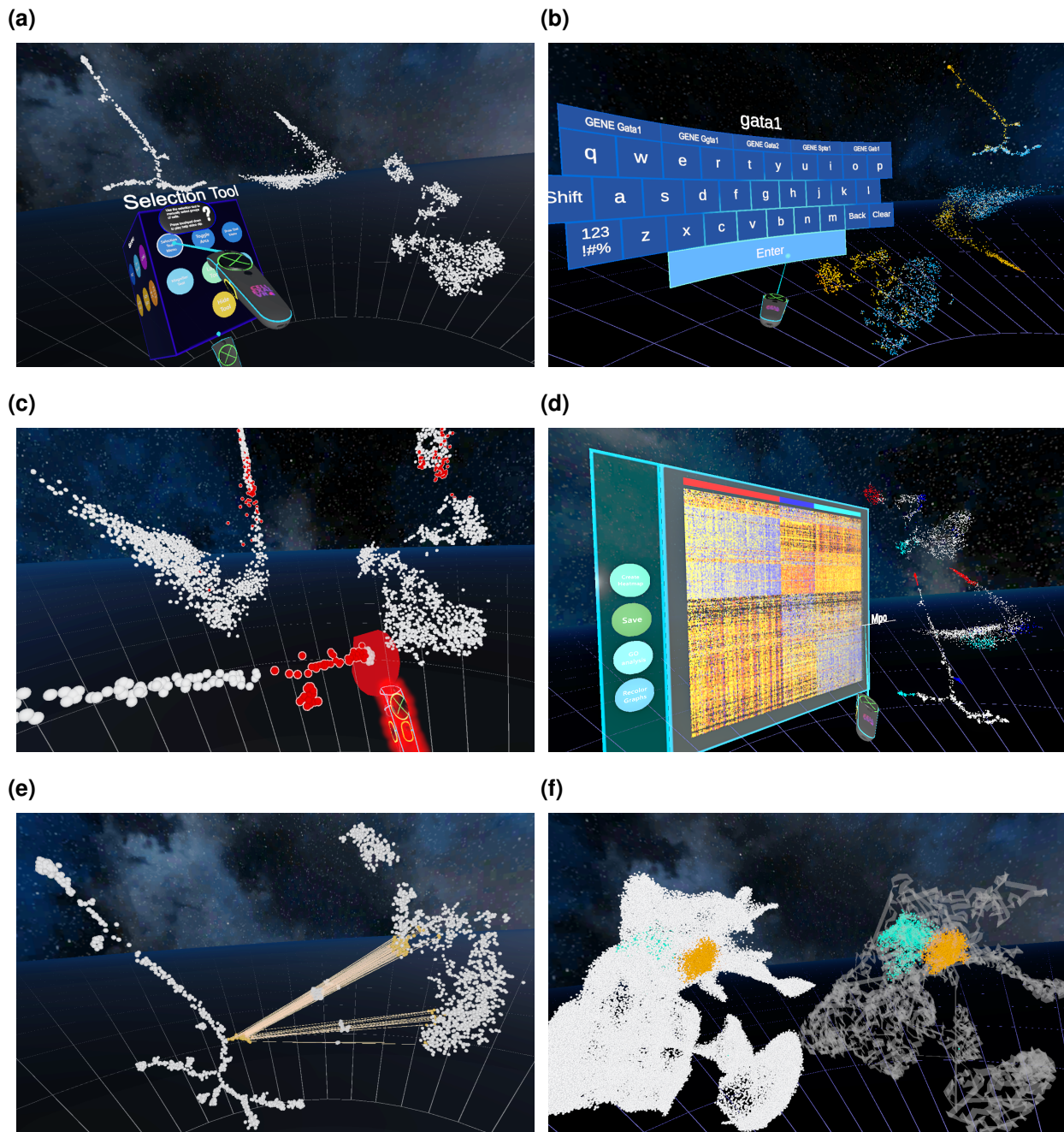
Figure 2: **(a)** Three sets of dimension reduced data (DDRTree, diffusion map, tSNE) from mouse hematopoietic stem and progenitor cells loaded into CellexalVR with the controllers in the foreground. **(b)** Cells in each DR plot coloured by the expression of Gata1 selected using the virtual keyboard. **(c)** The freehand selection tool. Passing cells through the paddle selects them into a group, and the corresponding cell in the other MDS plots are also highlighted simultaneously. Clicking the action-controller touchpad changes the colour of the selection tool to initiate a new group. **(d)** Differentially expressed genes between free-hand selected groups displayed as a heatmap. Coloured bars at the top of the heatmap show where the cells came from in the original selection. **(e)** Cells selected from the DDRTree plot (left) are tracked to their counterpart cells in the tSNE plot (right), highlighting the fact they fall into two further potential groups. **(f)** Cell types can be coloured and spawned in a different graph if they are difficult to see in dense DR plots. Here, two-types of mesoderm have been selected and rendered within a skeleton of the data (right) making them easier to see since they are not obscured by uncoloured cells (left).
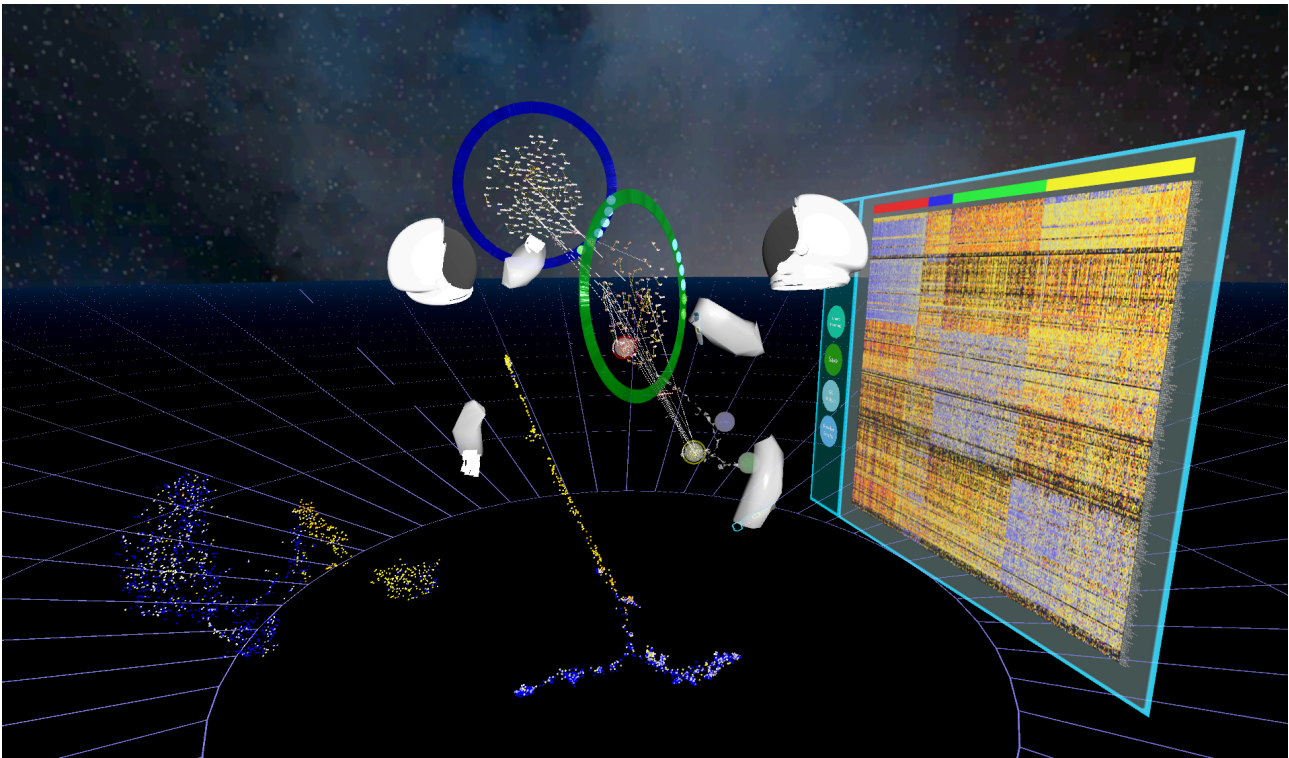
Figure 3: Multiuser mode showing two participants in the same CellexalVR session. Users can be in the same physical room using two separate workstations working between the same trackers, or in different locations on personal VR workstations. Other users can also be present in a session but not visible when in "ghost mode".

References

[1] E. Becht, L. McInnes, J. Healy, C. A. Dutertre, I. W. Kwok, L. G. Ng, F. Ginhoux, and E. W. Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 2019.

[2] A. Butler and R. Satija. Integrated analysis of single cell transcriptomic data across conditions, technologies, and species. *bioRxiv*, page 164889, 2017.

[3] L. Haghverdi, F. Buettner, and F. J. Theis. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*, 31(18):2989–2998, 2015.

[4] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 1982.

[5] S. Nestorowa, F. K. Hamey, B. Pijuan Sala, E. Diamanti, M. Shepherd, E. Laurenti, N. K. Wilson, D. G. Kent, and B. Göttgens. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood*, 128(8):e20–e31, 2016.

[6] B. Pijuan-Sala, J. A. Griffiths, C. Guibentif, T. W. Hiscock, W. Jawaid, F. J. Calero-Nieto, C. Mulas, X. Ibarra-Soria, R. C. V. Tyser, D. L. L. Ho, W. Reik, S. Srinivas, B. D. Simons, J. Nichols, J. C. Marioni, and B. Göttgens. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 2019.

[7] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, H. Clevers, B. Deplancke, I. Dunham, J. Eberwine, R. Eils, W. Enard, A. Farmer, L. Fugger, B. Göttgens, N. Hacohen, M. Haniffa, M. Hemberg, S. Kim, P. Klenerman, A. Kriegstein, E. Lein, S. Linnarsson, E. Lundberg, J. Lundeberg, P. Majumder, J. C. Marioni, M. Merad, M. Mhlanga, M. Nawijn, M. Netea, G. Nolan, D. Pe'er, A. Phillipakis, C. P. Ponting, S. Quake, W. Reik, O. Rozenblatt-Rosen, J. Sanes, R. Satija, T. N. Schumacher, A. Shalek, E. Shapiro, P. Sharma, J. W. Shin, O. Stegle, M. Stratton, M. J. Stubbington, F. J. Theis, M. Uhlen, A. Van Oudenaarden, A. Wagner, F. Watt, J. Weissman, B. Wold, R. Xavier, and N. Yosef. The human cell atlas. *eLife*, 6, 2017.

[8] M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9):865–868, 2017.

[9] V. Svensson, R. Vento-Tormo, and S. A. Teichmann. Exponential scaling of single-cell RNA-seq in the past decade. *Nature Protocols*, 13(4):599–604, 2018.

[10] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32(4):381–386, 2014.

[11] L. Van Der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[12] F. A. Wolf, P. Angerer, and F. J. Theis. SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, 2018.

Methods

CellexalVR is built using Unity 3D (https://unity3d.com/), an engine focusing mainly on video games. Unity handles tasks such as rendering the frames that are displayed on the computer's monitor and in the user's headset, collecting input from the keyboard and the controllers and forwarding events that trigger certain actions and handling all physics simulation. Unity comes with an editor which is the primary development environment that CellexalVR was created with. CellexalVR uses several Unity Assets, which are libraries containing scripts that handle parts of the program logic. SteamVR (https://steamcommunity.com/steamvr) and OpenVR (https://github.com/ValveSoftware/openvr) handles communication between the computer, headset and controllers and VRTK (https://github.com/thestonefox/VRTK) handles basic interaction logic such as the grabbing of objects.

Input files for CellexalVR are prepared in R using our cellexalvrR package (https://github.com/sonejilab/cellexalvrR). An example session is given below, but in brief one creates a cellexalvrR S4 object containing all relevant data which is then exported to correctly formatted files using the *export2cellexalvr* function.

Multi-user mode is facilitated via the Photon Unity Networking (PUN, https://www.photonengine.com/en-US/PUN) that works by sending packages that contain information about events between the users through Remote Procedure Calls (RPCs). The RPC ensure that each user's session is synchronised with all others and each person is looking at the same thing. Data is not transmitted over RPCs, therefore each user must have a copy of the data being analysed locally on the machine they are using. Head models were downloaded from NASA (https://nasa3d.arc.nasa.gov) and the arm models were made in Autodesk (https://www.autodesk.eu/).

In-session calculations are also performed in R using cellexalvrR. When a data set is loaded an R server is created and initialised with the expression data. Differentially expressed genes for heatmaps are calculated using the Wilcox test (implemented in C++ for speed) and clustering is performed using hierarchical clustering for the top *N* genes (default is 250) (Fig S5). Transcription factor networks within selected groups of cells can be calculated using the rho value (default) from the propr package, or partial correlations from the ppcor package. The top 130 interactions are returned by default, but is user configurable. All heatmaps and networks are rendered in the CellexalVR UI. TFs are defined as those in the AnimalTFDB database (urlhttp://bioinfo.life.hust.edu.cn/AnimalTFDB/).

Data formats. CellexalVR requires multiple correctly formatted files. This process is simplified by using the R package cellexalvrR that will generate the required files, R object, and SQLite database of expression data. At a minimum CellexalVR should be provided with:

- *A matrix of gene expression data (C cells x G genes)*. This is processed by cellexalvrR to an SQLite3 database that CellexalVR queries when needed.

- *At least one set of DR coordinates placing the cells in 3D space (C cells x 3)*. This can be from any DR methods the user deems suitable and CellexalVR will accept more than one DR table. These are exported as 4 column text files (*.drc) with the Cell ID in the first column.

In addition to these, further optional files can be imported. These are:

- *Surface marker intensities (C cells x S surface markers)*. These are recorded when cells have been index sorted. If using CITEseq, these expression values go into this table.

- *Cell type information (C cells x T types)*. These allow the user to label each cell as being of a certain type, which can then be displayed in the CellexalVR session. Cells are marked as belonging to a class with a "1", or "0" otherwise.

- *Metadata for cells (C cells x M meta)*. Further labels for cells, for example cell-cycle stage.

- ***Metadata for genes (G genes x M meta)***. For example, marking genes if they belong to a particular category such as transcription factors, epigenetic factors, or code surface proteins.

To export the necessary file from R, a cellexalvrR S4 object needs to be created first using the supplied functions . A typical export process would look as follows:

```
library(cellexalvrR)

#The following lines load the example data from Nesterowa et al (see references).

#Data can be downloaded from cellexalvr.med.lu.se

load("log2data.RData") #expression data (matrix)

load("facs.RData") #surface marker expression (matrix)

load("cell.ids.RData") #cell IDs (matrix)

load("diff.proj.RData") #diffusion map projection coordinates (matrix)

load("ddr.proj.RData") #DDRTree projection coordinates (matrix)

load("tsne.proj.RData") #tSNE projection coordinates (matrix)


log2data[1:10,1:10] #displays the first 10 rows and columns of the expression matrix


#The next 4 lines show how the matricies should look

head(facs)

head(cell.ids)


head(diff.proj)

head(ddr.proj)


#The 3 sets of MDS coordinates are put into a single list

proj.list <- list(diffusion=diff.proj,DDRtree=ddr.proj,tSNE=tsne.proj)

names(proj.list)


#Create a cellexalvr object setting the specie to mouse

cellvr <- MakeCellexaVRObj(log2data,mds.list=proj.list,specie="mouse",
```

```
cell.meta=cell.ids,facs.data=facs)


#Output the files to a selected folder.

export2cellexalvr(cellvr,"CellexalOut/")
```

More details on how to export CellexalVR ready files can be seen at https://www.cellexalvr.med.lu.se/cellexalvrr-vignette.

Hardware. CellexalVR was developed for HTC Vive(Pro) on a gaming class workstation comprising an Intel i7 processor, 16Gb RAM, 1Tb SSD, and an NVIDIA GTX1080 graphics card.