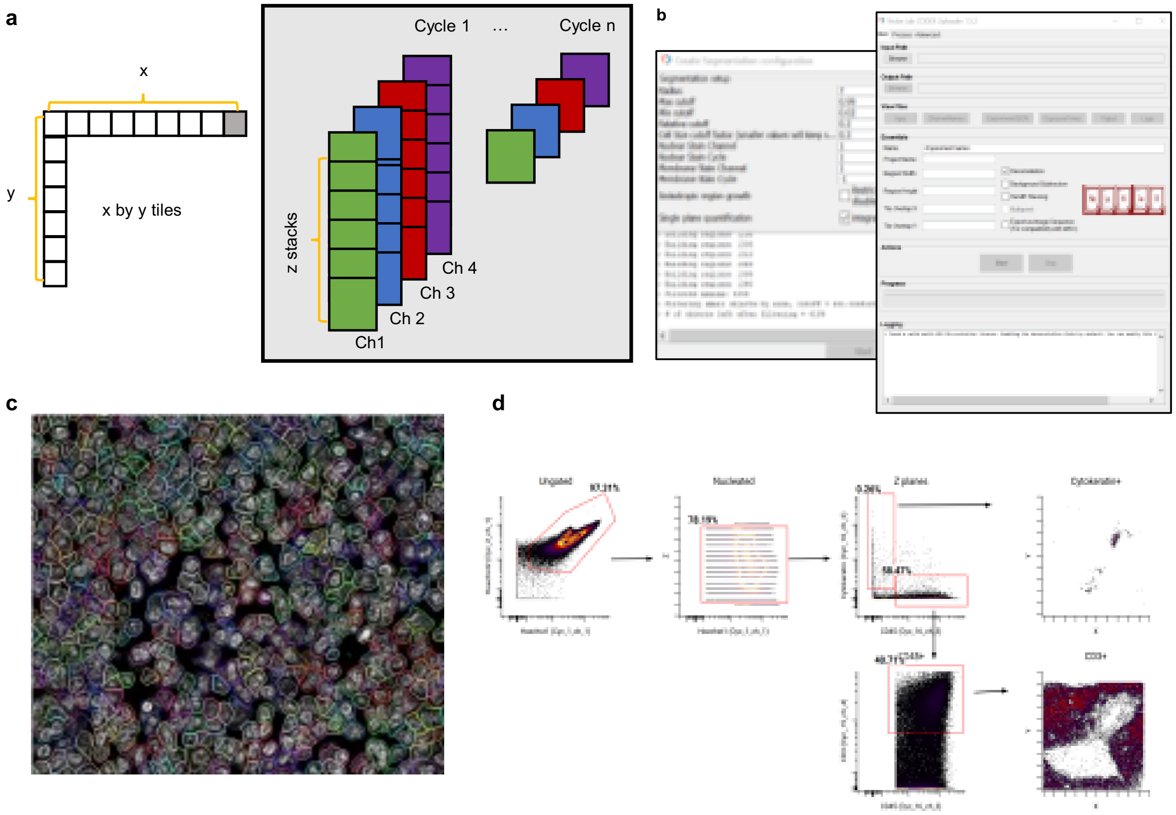
SUPPLEMENTARY INFORMATION

Extended Data Analysis Recommendations and Instructions

Supplementary Note 1: Image Data Overview

A typical CODEX dataset is composed of images broken down by the following dimensions (**Supplementary Fig. 1a**):

- **# of regions:** e.g., four different regions of tissue or four different samples
- Region size: e.g., 7x9 tiles (with 30% overlap)
- **# of Z stacks:** e.g., 15 (at 1.5 µm each), which ensures in-focus image capture across a large tissue region
- **# of channels**: e.g., CH1: Hoechst, CH2: Alexa488, CH3: ATTO550, CH4: Alexa647
- **# of cycles:** e.g., Cycle 3: Hoechst, CD3, CD4, CD8; Cycle 4: Hoechst, CD31, CD19, podoplanin.



Supplementary Figure 1. CODEX data analysis pipeline. a, A visual representation of the data structure for CODEX imaging files. **b**, The user interfaces for the CODEX Uploader and Segmentation software. **c**, A visual representation of the single-cell segmentation masks for a CODEX image. **d**, Key steps for manual cleaning the segmented data using a flow cytometry software platform.

Data will be generated from the scope as individual images across all dimensions. These images need to be concatenated to form a 7x9 imaging region, contain all z stacks, and all channels for each cycle. There is minor drift of the microscope in between cycles, so drift compensation must then be done to align markers across all channels. This is done with the Hoechst or DRAQ5 (nuclear staining) used for all cycles as a reference.

Once all dimensions have been merged, it is necessary to implement cell segmentation to quantitatively describe the image in single-cell resolution. This results in a file with each row representing a segmented cell and each column indicating a feature of that cell (e.g., X/Y positions, Channel intensities, original tile number).

Supplementary Note 2: Image Processing

To automate and standardize these image processing steps, our laboratory has published an open source software, CODEXSetup (available at https://github.com/nolanlab/CODEX). There is a single software interface for combining image dimensions and drift compensation (CODEXUploader.exe) and another for segmentation (CODEXSegm.exe). Within CODEXUploader the necessary parameters that must be set are input and output directories and X and Y region size (**Supplementary Fig. 1b**). Other parameters should be auto-populated based on the Experiment.json, channelnames.txt, and expsouretimes.txt files found in the directory with your images. There are additional options for background subtraction (blank cycle), including processing of H&E staining (if completed), processing for multipoint imaging (especially useful for tissue microarrays), deconvolution (if there is a microevolution license), and exporting as MAV sequence (downstream analysis with Akoya Biosciences's ImageJ MAV plugin). Finally, there is an option for previewing the processing to make sure that parameters used are appropriate; this is important as the image processing can be time consuming.

Once run, the program will produce the following outputs in the specified output directory:

- Concatenated individual tiles across all dimensions
- bestFocus folder with:
    - Concatenated individual tiles with best focal plane selected
    - Montages of stitched concatenated individual tiles with best focal planes selected (**Supplementary Fig. 1a**)

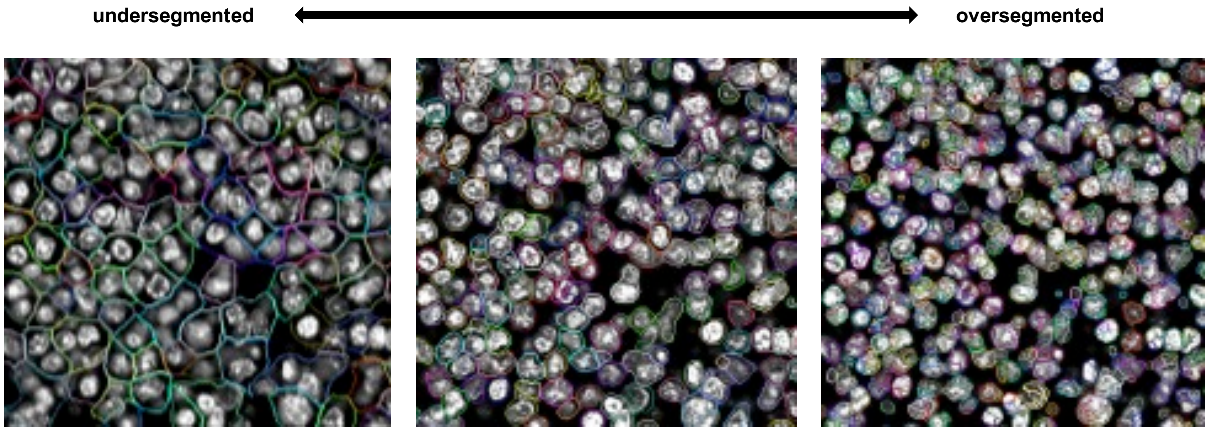
Supplementary Note 3: Single-cell Segmentation

After initial data processing, segmentation can be run on the output folder generated. Within CODEXSegm the following parameters must be set based on the dataset for volumetric watershed segmentation (good starting point values are given in parentheses) (**Supplementary Fig. 1c**):

- Radius: The estimated radius of nuclei size (6)
- maxCutoff: The highest signal to be considered a nuclei to filter potential debris (0.99)
- minCutoff: The lowest signal to be considered a part of a nuclei (0.07)
- relativeCutoff – Used for region growth relative to peak intensity of each nuclei; usually not changed (0.2)
- sizeCutOffFactor – Filters cells smaller than expected cell size based on radius (0.5)
- Nuclear stain channel: The channel number of nuclear stain

- Nuclear stain cycle: The number of the nuclear stain cycle
- Membrane stain channel: The channel number of membrane stain
- Membrane stain cycle: The number of the membrane stain cycle; if there is no membrane stain, then segmentation can be done with nuclear channel only and the number is -1
- Anisotropic region growth: If checked, this restricts pos-neg separation between crowded cell types (typically not checked)
- Single Plane Segmentation: If checked, this integrates the signal only based on one plane instead of volumetrically (typically checked)

CODEXSegm also provides an option to preview the segmentation parameters on one tile. It is usually an iterative process to optimize the segmentation mask for representative regions of the tissue. It is generally better to oversegment than to undersegment because single cell data will get extracted from within the mask. Undersegmentation will include multiple cells in the same mask and thus cells will appear “positive” for both cell-type markers confounding the cell-type identification results (**Supplementary** Fig. 2). Oversegmentation will occur when multiple masks are drawn through one cell, while not as problematic for cell-type identification, this will pose problems for cell frequency assessments. Optimal settings would occur where each single cell is individually segmented with its own mask.



Supplementary Figure 2. Demonstration of poor and good segmentation results. Undersegmented, optimally segmented, and oversegmented images are represented with masks overlaid on one tile from (Hoechst nuclear staining) from the multicycle.

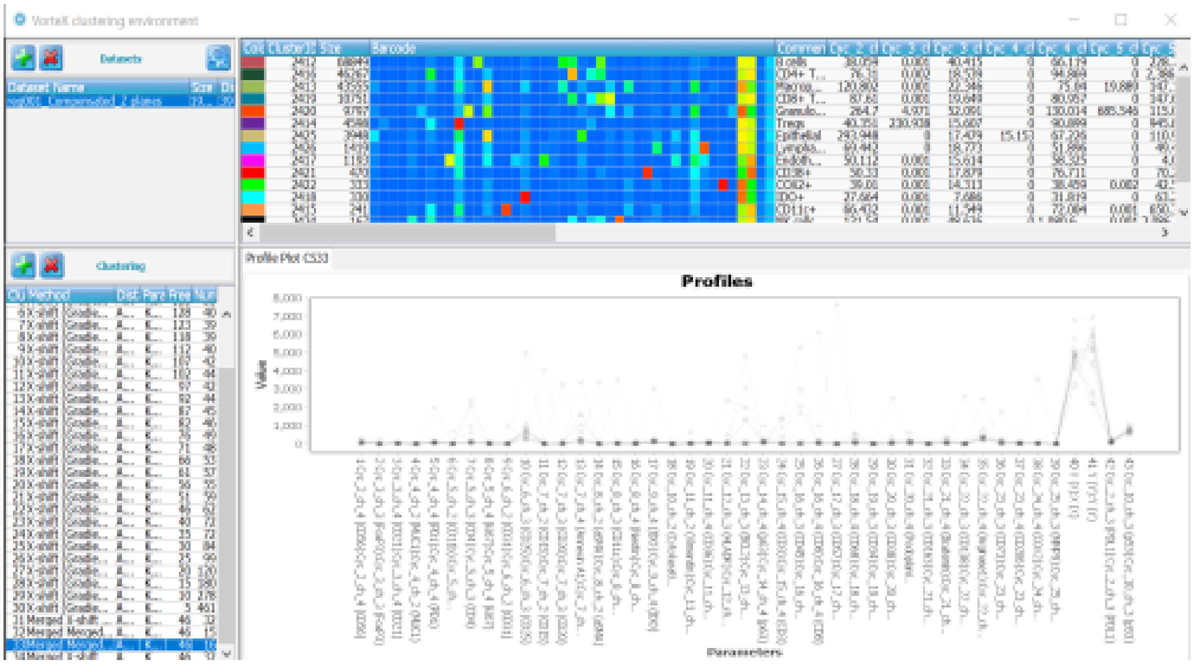
Once run, the program will produce the following outputs in the directory where segmentation was performed:

- Dataframe for each region using either compensated or uncompensated settings that include channel intensities and location for each segmented cell
    - FCS and CSV formats are generated
- Individual tile segmentation masks
    - TXT and PNG formats are generated

Supplementary Note 4: Cell type Identification

The individual data frames can be processed and analyzed with methods used for other single-cell techniques like flow cytometry and mass cytometry. Cell-type identification can be performed by manual gating or unsupervised cell type clustering.

First, it is necessary to import compensated files into a manual gating program like CellEngine or FlowJo. This allows selection of nucleated cells by gating for dual-positive nucleated cells (**Supplementary Fig. 3**). Selection of cells from a best focal plane not on the extremes of the Z stacks is necessary to eliminate noise in the data prior to unsupervised clustering.



Dataset Name	Size	Di
reg001_Compensated_Z planes	19...	20

Col	Cluster ID	Size	Barcode	Comment	Cyc_2_c	Cyc_3_c	Cyc_3_c	Cyc_4_c	Cyc_4_c	Cyc_5_c	Cyc_5
🟥	2412	68849		B cells	39.054	0.001	40.415	0	95.173	0	278.
■	2416	46262		CD4+ T...	76.31	0.001	18.539	0	94.869	0	2,386
🟩	2413	43555		Macrop...	120.802	0.001	22.346	0	75.14	19.889	147.
■	2419	10751		CD8+ T...	87.61	0.001	19.649	0	80.957	0	147.
🟧	2420	9797		Granulo...	264.7	4.971	52.091	0	130.014	685.546	115.
🟪	2414	4598		Tregs	40.351	230.938	15.607	0	90.899	0	945.
🟨	2425	3948		Epithelial	293.948	0	17.479	15.153	67.226	0	110.
🟦	2426	1419		Lympho...	69.442	0	18.773	0	51.896	0	49.
🟪	2417	1153		Endoth...	50.112	0.001	15.614	0	58.325	0	4.
🟥	2421	470		CD38+	50.33	0.001	17.879	0	76.711	0	70.
🟩	2422	333		CD82+	39.01	0.001	14.313	0	38.459	0.002	42.
🟦	2418	330		IDO+	27.664	0.001	7.686	0	31.819	0	61.
🟧	2415	241		CD11c+	66.432	0.001	11.549	0	72.084	0.001	630.

Clu	Method		Dist.	Para	Free	Num
6	X-shift	Gradie...	A...	K...	128	40
7	X-shift	Gradie...	A...	K...	123	39
8	X-shift	Gradie...	A...	K...	118	39
9	X-shift	Gradie...	A...	K...	112	40
10	X-shift	Gradie...	A...	K...	107	42
11	X-shift	Gradie...	A...	K...	102	42
12	X-shift	Gradie...	A...	K...	97	43
13	X-shift	Gradie...	A...	K...	92	44
14	X-shift	Gradie...	A...	K...	87	45
15	X-shift	Gradie...	A...	K...	82	46
16	X-shift	Gradie...	A...	K...	76	46
17	X-shift	Gradie...	A...	K...	71	48
18	X-shift	Gradie...	A...	K...	66	53
19	X-shift	Gradie...	A...	K...	61	57
20	X-shift	Gradie...	A...	K...	56	55
21	X-shift	Gradie...	A...	K...	51	58
22	X-shift	Gradie...	A...	K...	45	[illegible]
23	X-shift	Gradie...	A...	K...	40	70
24	X-shift	Gradie...	A...	K...	35	72
25	X-shift	Gradie...	A...	K...	30	84
26	X-shift	Gradie...	A...	K...	25	98
27	X-shift	Gradie...	A...	K...	20	120
28	X-shift	Gradie...	A...	K...	15	180
29	X-shift	Gradie...	A...	K...	10	278
30	X-shift	Gradie...	A...	K...	5	481
31	Merged	X-shift ...	A...	K...	46	32
32	Merged	Merged ...	A...	K...	46	15
33	Merged	Merged ...	A...	K...	46	16
34	Merged	X-shift	A...	K...	46	[illegible]

Supplementary Figure 3. Unsupervised single-cell clustering software. The user interface for VorteX clustering.

The cleaned datasets can be imported into Vortex, which is another unsupervised clustering (available at https://github.com/nolanlab/vortex/wiki/Getting-Started) (28). The data are imported into the clustering database. Markers that should be included for clustering (blue), those not used for clustering but kept for analysis (yellow), and markers to be removed (gray) are selected by clicking and changing the color. Parameters are the following (good starting points for CODEX data are given):

- Numerical transformation: none
- Noise threshold: uncheck
- Feature rescaling: none

- Normalization: none
- Minimal Euclidean length: uncheck
- Distance Measure: Angular Distance
- Clustering Algorithm: X-shift
- Density Estimate: N nearest neighbors (fast)
- Num. Neighbors for mode finding N: Determine Automatically

The program calculates clusters for each *K* between the default parameters. This can be used to find the elbow point for optimal clustering results. Alternatively, one can select to over-cluster by choosing a low *K* with a high corresponding cluster number and then manually merge clusters. Using the cluster investigation interface with expression profiles, minimal spanning trees, and divisive marker trees, cell type annotation can be done for each cluster based on marker expression. Finally, spatial locations of these clusters should be verified to ensure accuracy of the clustering.