

# scanpy.pp.filter\_genes

**scanpy.pp.filter\_genes**(*data*, *min\_counts=None*, *min\_cells=None*, *max\_counts=None*, *max\_cells=None*, *inplace=True*, *copy=False*)

Filter genes based on number of cells or counts.

Keep genes that have at least `min_counts` counts or are expressed in at least `min_cells` cells or have at most `max_counts` counts or are expressed in at most `max_cells` cells.

Only provide one of the optional parameters `min_counts`, `min_cells`, `max_counts`, `max_cells` per call.

## Parameters:

**data** : `AnnData`

An annotated data matrix of shape `n_obs` × `n_vars`. Rows correspond to cells and columns to genes.

**min\_counts** : `optional [int]` (default: `None`)

Minimum number of counts required for a gene to pass filtering.

**min\_cells** : `optional [int]` (default: `None`)

Minimum number of cells expressed required for a gene to pass filtering.

**max\_counts** : `optional [int]` (default: `None`)

Maximum number of counts required for a gene to pass filtering.

**max\_cells** : `optional [int]` (default: `None`)

Maximum number of cells expressed required for a gene to pass filtering.

**inplace** : `bool` (default: `True`)

Perform computation inplace or return result.

## Return type:

`Union [AnnData, None, Tuple [ndarray, ndarray]]`

## Returns:

: Depending on `inplace`, returns the following arrays or directly subsets and annotates the data matrix

gene\_subset

Boolean index mask that does filtering. `True` means that the gene is kept. `False` means the gene is removed.

---

## number\_per\_gene

Depending on what was thresholded ( `counts` or `cells` ), the array stores `n_counts` or `n_cells` per gene.