scanpy.pp.highly_variable_genes

scanpy.pp.highly_variable_genes(*adata*, *layer=None*, *n_top_genes=None*, *min_disp=0.5*, *max_disp=inf*, *min_mean=0.0125*, *max_mean=3*, *span=0.3*, *n_bins=20*, *flavor='seurat'*, *subset=False*, *inplace=True*, *batch_key=None*, *check_values=True*)

Annotate highly variable genes [Satija15] [Zheng17] [Stuart19].

Expects logarithmized data, except when `flavor='seurat_v3'`, in which count data is expected.

Depending on `flavor`, this reproduces the R-implementations of Seurat [Satija15], Cell Ranger [Zheng17], and Seurat v3 [Stuart19].

For the dispersion-based methods ([Satija15] and [Zheng17]), the normalized dispersion is obtained by scaling with the mean and standard deviation of the dispersions for genes falling into a given bin for mean expression of genes. This means that for each bin of mean expression, highly variable genes are selected.

For [Stuart19], a normalized variance for each gene is computed. First, the data are standardized (i.e., z-score normalization per feature) with a regularized standard deviation. Next, the normalized variance is computed as the variance of each gene after the transformation. Genes are ranked by the normalized variance.

See also `scanpy.experimental.pp._highly_variable_genes` for additional flavours (e.g. Pearson residuals).

Parameters:

adata : `AnnData`

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

layer : `Optional` [`str`] (default: `None`)

If provided, use `adata.layers[layer]` for expression values instead of `adata.X`.

n_top_genes : `Optional` [`int`] (default: `None`)

Number of highly-variable genes to keep. Mandatory if `flavor='seurat_v3'`.

min_mean : `Optional` [`float`] (default: `0.0125`)

If `n_top_genes` unequals `None`, this and all other cutoffs for the means and the normalized dispersions are ignored. Ignored if `flavor='seurat_v3'`.

---

max_mean : **`Optional`**[**`float`**] (default: `3`)

If `n_top_genes` unequals `None`, this and all other cutoffs for the means and the normalized dispersions are ignored. Ignored if `flavor='seurat_v3'`.

---

min_disp : **`Optional`**[**`float`**] (default: `0.5`)

If `n_top_genes` unequals `None`, this and all other cutoffs for the means and the normalized dispersions are ignored. Ignored if `flavor='seurat_v3'`.

---

max_disp : **`Optional`**[**`float`**] (default: `inf`)

If `n_top_genes` unequals `None`, this and all other cutoffs for the means and the normalized dispersions are ignored. Ignored if `flavor='seurat_v3'`.

---

span : **`Optional`**[**`float`**] (default: `0.3`)

The fraction of the data (cells) used when estimating the variance in the loess model fit if `flavor='seurat_v3'`.

---

n_bins : **`int`** (default: `20`)

Number of bins for binning the mean gene expression. Normalization is done with respect to each bin. If just a single gene falls into a bin, the normalized dispersion is artificially set to 1. You'll be informed about this if you set `settings.verbosity = 4`.

---

flavor : **`Literal`**[`'seurat'`, `'cell_ranger'`, `'seurat_v3'`] (default: `'seurat'`)

Choose the flavor for identifying highly variable genes. For the dispersion based methods in their default workflows, Seurat passes the cutoffs whereas Cell Ranger passes `n_top_genes`.

---

subset : **`bool`** (default: `False`)

Inplace subset to highly-variable genes if `True` otherwise merely indicate highly variable genes.

---

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

Whether to place calculated metrics in `.var` or return them.

---

batch_key : `Optional` [`str`] (default: `None`)

If specified, highly-variable genes are selected within each batch separately and merged. This simple process avoids the selection of batch-specific genes and acts as a lightweight batch correction method. For all flavors, genes are first sorted by how many batches they are a HVG. For dispersion-based flavors ties are broken by normalized dispersion. If `flavor = 'seurat_v3'`, ties are broken by the median (across batches) rank based on within-batch normalized variance.

check_values : `bool` (default: `True`)

Check if counts in selected layer are integers. A Warning is returned if set to True. Only used if `flavor='seurat_v3'`.

Return type: `Optional` [`DataFrame`]

Returns:

: Depending on `inplace` returns calculated metrics (`DataFrame`) or updates `.var` with the following fields

highly_variable : `bool`

boolean indicator of highly-variable genes

means

means per gene

dispersions

For dispersion-based flavors, dispersions per gene

dispersions_norm

For dispersion-based flavors, normalized dispersions per gene

variances

For `flavor='seurat_v3'`, variance per gene

variances_norm

For `flavor='seurat_v3'`, normalized variance per gene, averaged in the case of multiple batches

highly_variable_rank : `float`

For `flavor='seurat_v3'`, rank of the gene according to normalized variance, median rank in the case of multiple batches

highly_variable_nbatches : `int`

If batch_key is given, this denotes in how many batches genes are detected as HVG

highly_variable_intersection : `bool`

If batch_key is given, this denotes the genes that are highly variable in all batches

Notes

This function replaces `filter_genes_dispersion()`.