Analysis of scRNA sequencing data of human brain cells

Anastasia Popova

Outline

01 Introduction

02 Raw Data Processing

03 Exploratory Data Analysis

04 Data Preparation: Filtering, Normalization, Dimensionality reduction

05 Clusterization

Introduction

motivation

Understanding which specific proteins are responsible for the development of autoimmune diseases in humans can aid in early diagnosis and treatment.

One of the most progressive research methods today is **single-cell RNA sequencing** (scRNA-seq), which provides a detailed and accurate picture of gene expression.

goal

In this project, I will analyze single-cell RNA sequencing data from the human brain. The studied cells were obtained from different patients with various autoimmune diseases.

method

I will use the **augmented transcriptome mapping** of RNA sequencing data, a fast and highly resolved method for analyzing gene expression [1].

Because RNA-seq data analysis mainly relies on packages for R language, here I also aim to entirely **rely on Python**. The main reason for this is the belief that big data processing can be more effective using Python. It is also a more flexible and fastly developing tool, using which also might be crucial for cutting-edge data analysis.

[1] Workflow is mainly based on recommendations in https://www.sc-best-practices.org/ book.

Raw Data Processing

steps of raw data processing

1. Generation so-called the **splici index** (spliced transcripts + introns) for the reference genome (using pyroe [1] for Genome sequence, primary assembly (GRCh38) and Comprehensive gene annotation (CHR) from Genecode) and index it (using salmon [2])

2. Generation a **permit list** for cell barcode correction 10x Chromium v2 chemistry

3. **Mapping** of the sequencing reads against an index of the reference file (alevin-fry [3])

4. **UMI resolution**: allocation a molecular count to each gene in each cell. UMIs with equal matches to multiple genes are treated as a group (a.k.a. equivalence class), using an expectation maximization algorithm (alevin-fry [3]).

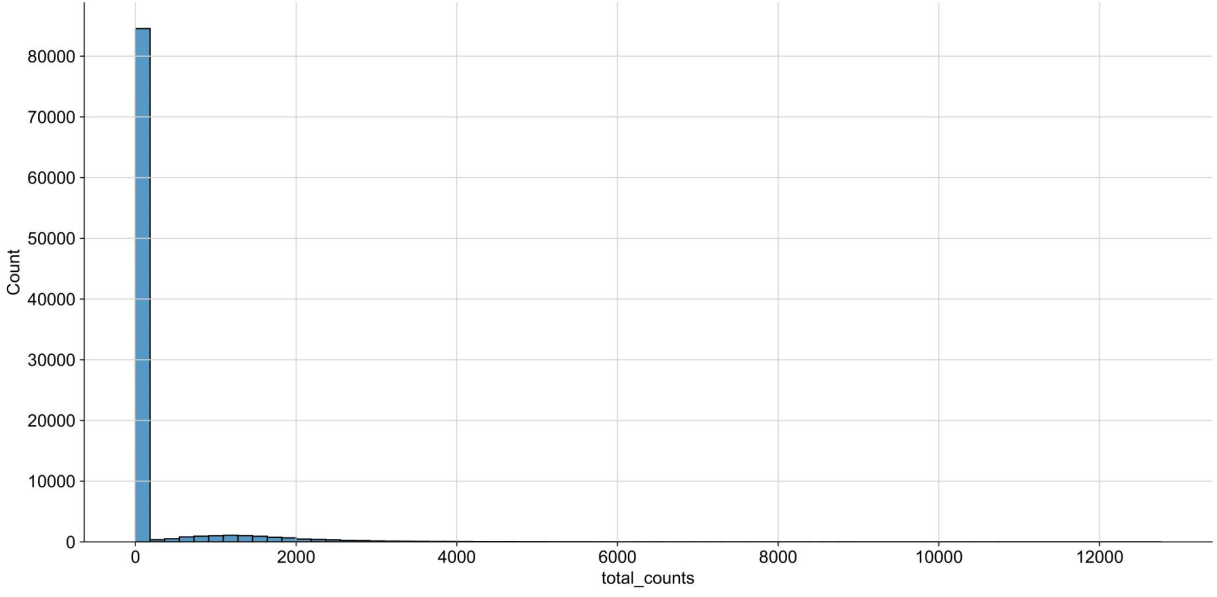
[1]https://pyroe.readthedocs.io/en/latest/

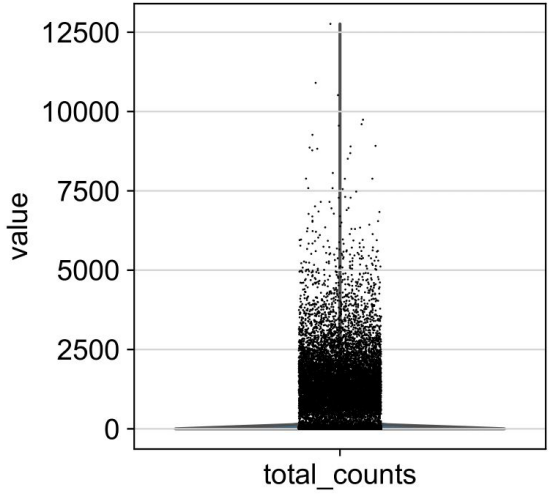
[2]https://combine-lab.github.io/salmon/

[3]https://alevin-fry.readthedocs.io/en/latest/

Exploratory Data Analysis

After raw data processing, the dataset consists of **95,383 cells and 62,700 genes**. The number of unique gene names is 61,228 and gene names contain digits, for example, 'RNVU1-22', and 'RNVU1-3'.





Here and further scanpy and anndata Python libraries were used.

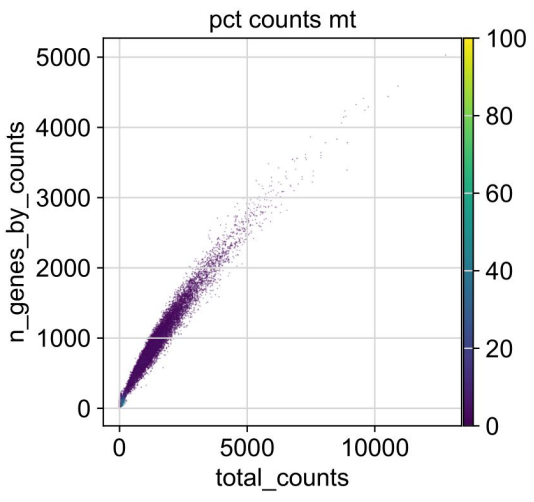
Filtering out ribosomal and mitochondrial counts

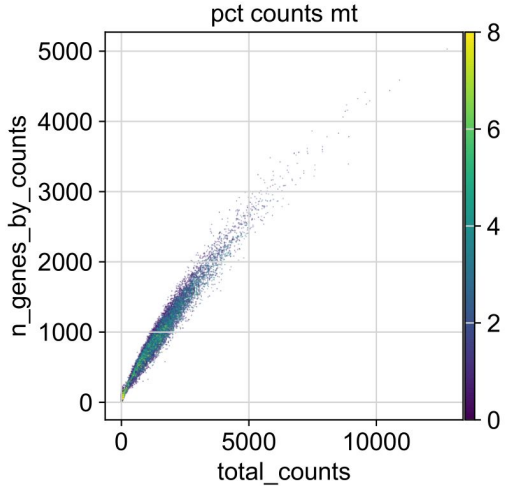
To filter out cells with broken membranes, which usually have a high amount of mitochondrial and ribosomal counts, few detected genes and a low count depth (the number of counts per barcode).

A quality control threshold: outliers are data points that differ by **5 mean absolute deviations**. Also, cells with a percentage of mitochondrial or ribosomal counts exceeding **8** % are filtered out.

- Number of mitochondrial genes: 37; Number of ribosomal genes: 1,715
- Number of cells after filtering of low quality cells: 92,834

before

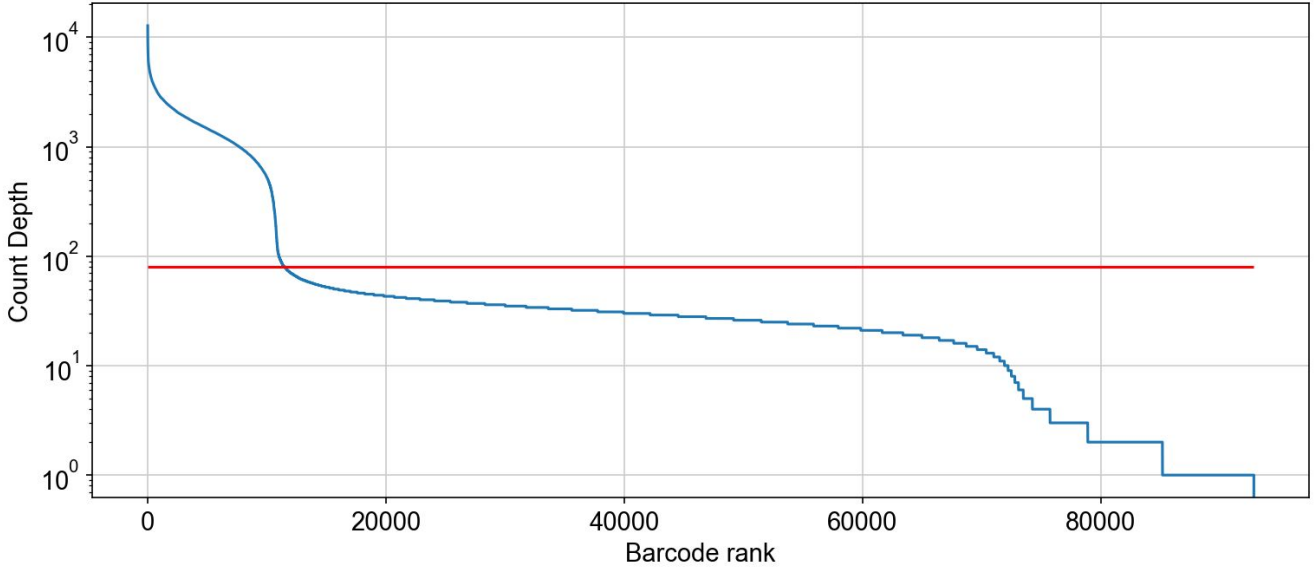




after

Ambient RNA Correction

Droplet-based scRNA-seq protocol assumes that UMI enables to identify of the number of molecules for each gene and each cell. But cell-free mRNA molecules can be present in the dilution (people refer to this as a "soup"), which are also sequenced. Here cell-free (ambient) mRNA are filtered out for **count depth < 80**.



Total number of cells: 92,834

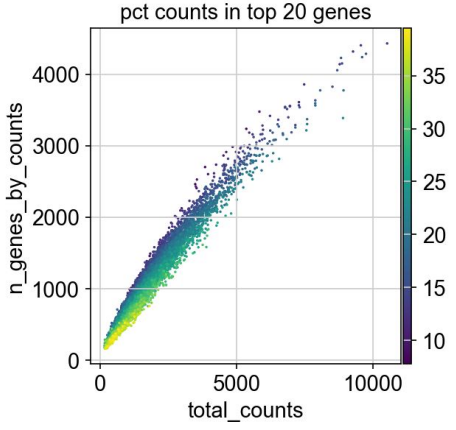
Number of cells after filtering of low quality cells: 11,444

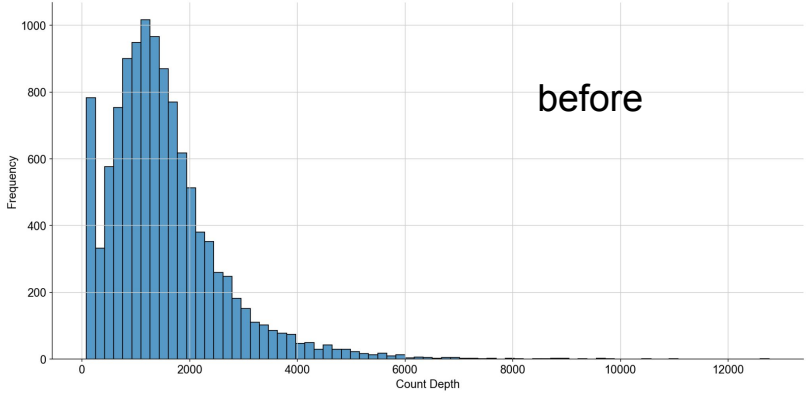
Filtering Outliers

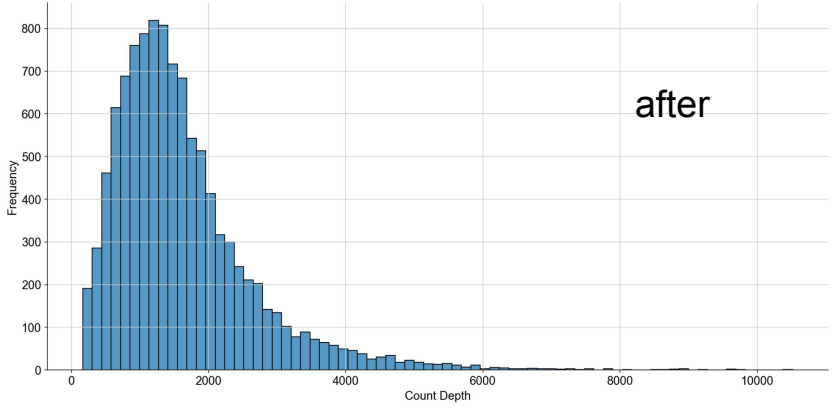
As for mitochondrial or ribosomal counts, we filter out low-quality counts for the whole dataset.

A quality control threshold: outliers are data points that differ by 5 mean absolute deviations.

Number of cells after filtering of low-quality cells: 10,703



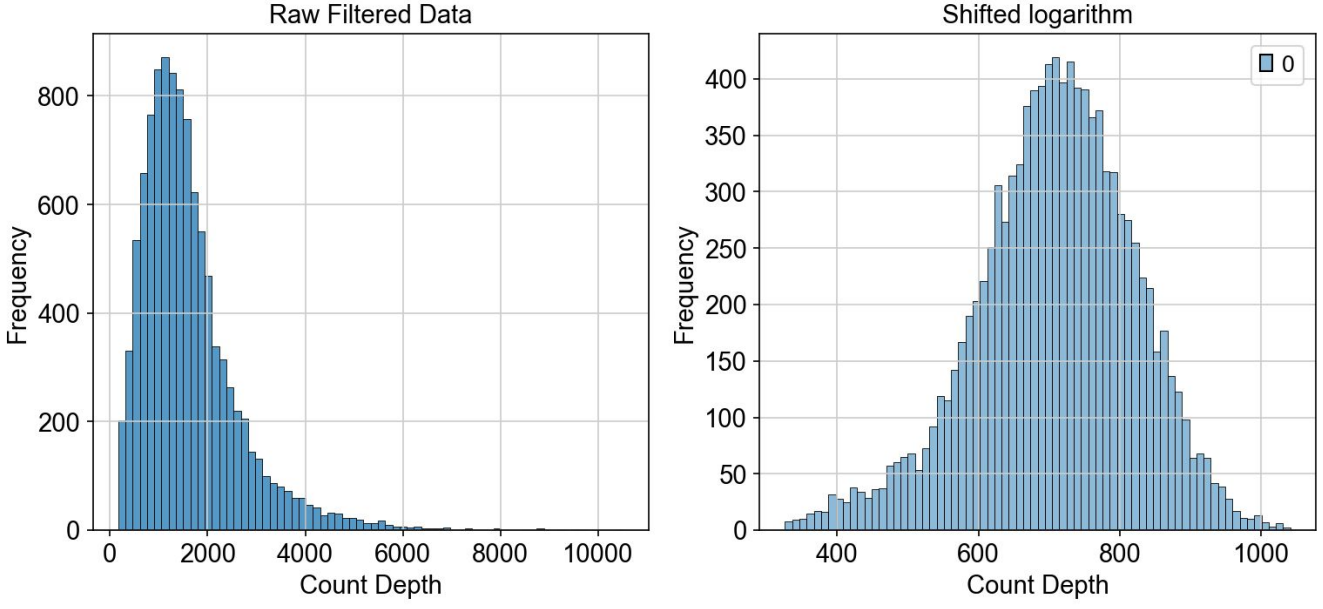




Normalization

One of the fastest ways to normalize data is by applying **shifted logarithm transformation**.

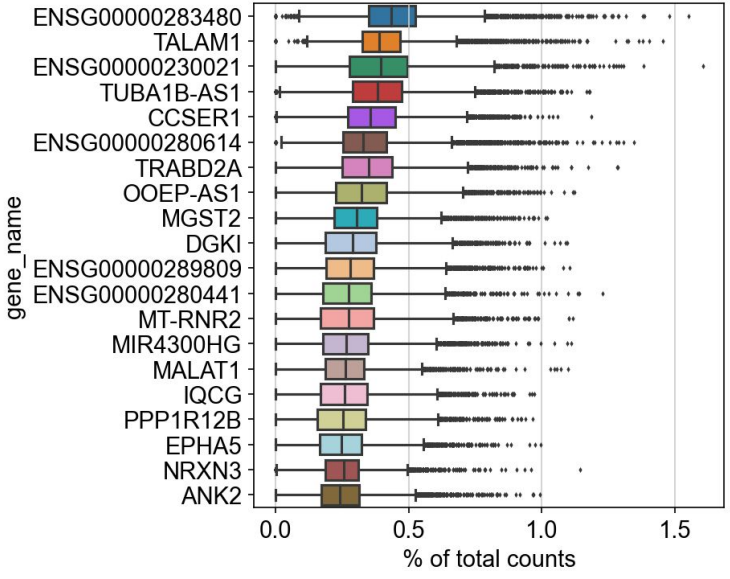
It aims to make the variances across the dataset more homogenous. It is used for subsequent dimensionality reduction and identification of differentially expressed genes.

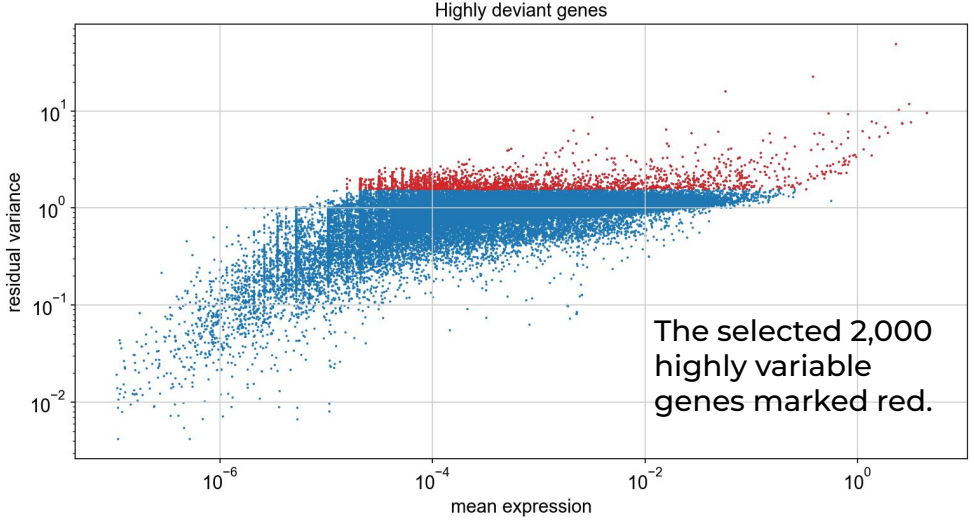


Gene Selection

Some number of genes might be detected in a few cells (the matrix is sparse), but usually, it is interesting to consider the most expressed ones and genes with high spread of expression values relative to the mean.

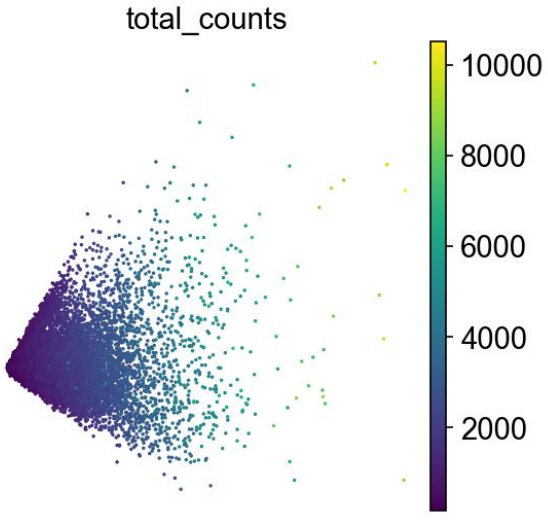
Analytic Pearson residuals help detect how much each gene deviates from the constant-expression model [1].



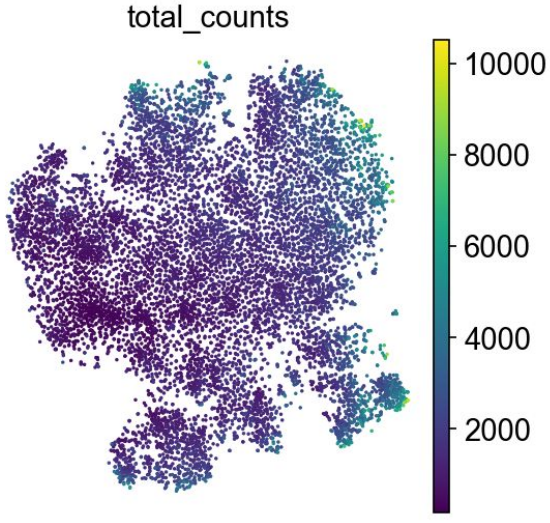


[1] For more information and comparisons to other gene selection methods, refer to Lause et al. (2021).

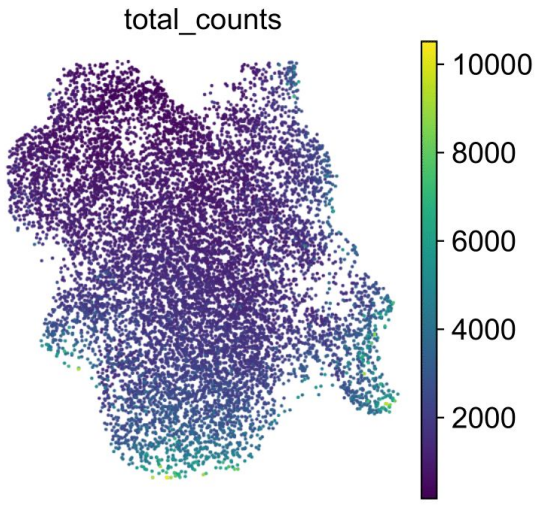
Dimensionality Reduction



PCA

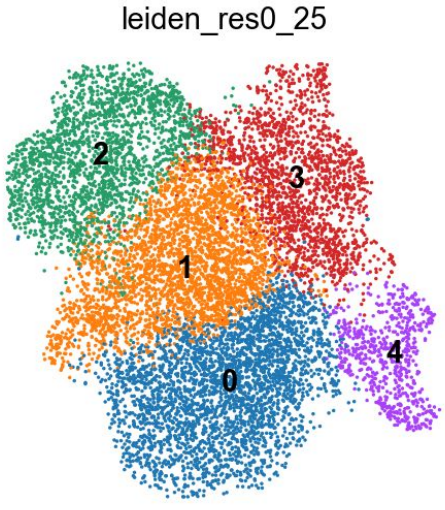


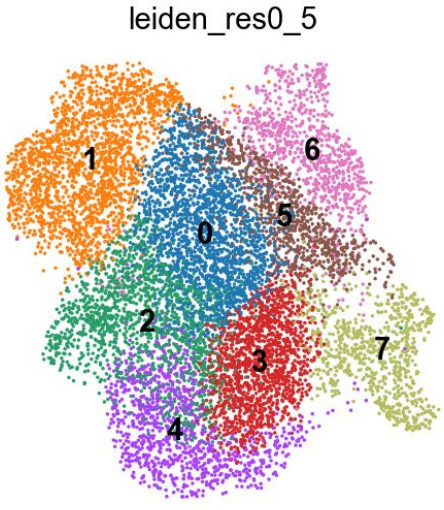
t-SNE

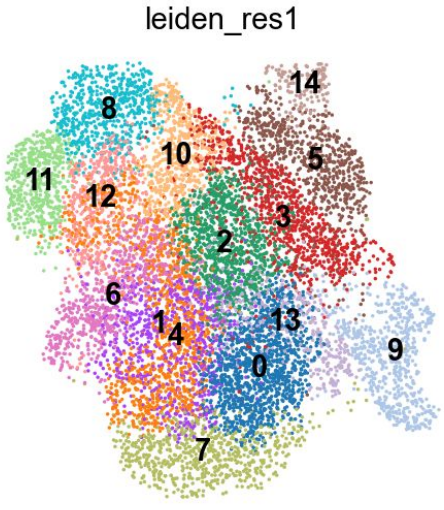


UMAP

Clusterization







The **Leiden method** works by optimizing the modularity score, which measures the strength of connections within communities compared to connections between communities. By iteratively reassigning nodes to different communities, the method aims to find the arrangement that maximizes modularity, revealing distinct groups within the network.