baredSC

Release 1.0.0

Jean-Baptiste Delisle, Lucille Lopez-Delisle

Jun 04, 2021

CONTENTS:

CHAPTER

ONE

BARED (BAYESIAN APPROACH TO RETREIVE EXPRESSION DISTRIBUTION OF) SINGLE CELL

baredSC is a tool that uses a Monte-Carlo Markov Chain to estimate a confidence interval on the probability density function (PDF) of expression of one or two genes from single-cell RNA-seq data. It uses the raw counts and the total number of UMI for each cell. The PDF is approximated by a number of 1d or 2d gaussians provided by the user. The likelihood is estimated using the asumption that the raw counts follow a Poisson distribution of parameter equal to the proportion of mRNA for the gene in the cell multiplied by the total number of UMI identified in this cell.

1.1 Installation

1.1.1 Requirements

It as only been tested on linux but should work on MacOS.

It requires python >= 3.7 (tested 3.7.3 and 3.9.1)

Dependencies of classical python packages:

- numpy (tested 1.16.4 and 1.19.5)
- matplotlib (tested 3.1.1 and 3.3.4)
- pandas (tested 0.25.0 and 1.2.1)
- scipy (tested 1.3.0 and 1.6.0)

Dependencies of a python package from Jean-Baptiste Delisle dedicated to mcmc:

- samsam (above 0.1.2)

1.1.2 Installation

For the moment you can install it with pip:

pip install –extra-index-url https://obswww.unige.ch/~delisle baredSC

It may be accessible in conda later.

1.2 Usage

– *Optional arguments to evaluate logevidence*

1.2.1 General usage

Here is a description of all possible parameters. The tool take around 1 minute for 1d, 2000 cells, default parameters, independently of the number of gaussian and around 30 seconds for 300 cells. For the 2d 300 cells is around 3 minutes, 2000 cells around 15 minutes. Increasing the number of samples in the MCMC or in the burning phase will increase the time.

1.2.2 baredSC_1d

Run mcmc to get the pdf for a given gene using a normal distributions.

```
usage: baredSC_1d [-h] --input INPUT --geneColName GENECOLNAME
                  [--metadata1ColName METADATA1COLNAME]
                  [--metadata1Values METADATA1VALUES]
                  [--metadata2ColName METADATA2COLNAME]
                  [--metadata2Values METADATA2VALUES]
                  [--metadata3ColName METADATA3COLNAME]
                  [--metadata3Values METADATA3VALUES] [--xmin XMIN]
                  [--xmax XMAX] [--xscale {Seurat,log}]
                  [--targetSum TARGETSUM] [--nx NX] [--osampx OSAMPX]
                  [--osampxpdf OSAMPXPDF] [--minScale MINSCALE]
                  [--nnorm NNORM] [--nsampMCMC NSAMPMCMC]
                  [--nsampBurnMCMC NSAMPBURNMCMC]
                  [--nsplitBurnMCMC NSPLITBURNMCMC] [--T0BurnMCMC T0BURNMCMC]
                  [--seed SEED] [--minNeff MINNEFF] [--force] --output OUTPUT
                  [--figure FIGURE] [--title TITLE]
                  [--removeFirstSamples REMOVEFIRSTSAMPLES]
                  [--nsampInPlot NSAMPINPLOT] [--prettyBins PRETTYBINS]
                  [--logevidence LOGEVIDENCE] [--coviscale COVISCALE]
                  [--nis NIS] [--version]
```

Named Arguments

--version show program's version number and exit

Required arguments

--input Input table with one line per cell columns with raw counts and one column nCount_RNA with total number of UMI per cell optionally other meta data to filter.

--geneColName Name of the column with gene counts.

--output Ouput file basename (will be npz) with results of mcmc.

Optional arguments to select input data

--metadata1ColName Name of the column with metadata1 to filter.

--metadata1Values Comma separated values for metadata1.

--metadata2ColName Name of the column with metadata2 to filter.

--metadata2Values Comma separated values for metadata2.

--metadata3ColName Name of the column with metadata3 to filter.

--metadata3Values Comma separated values for metadata3.

Optional arguments to run MCMC

--xmin Minimum value to consider in x axis.

Default: 0

--xmax Maximum value to consider in x axis.

Default: 2.5

--xscale Possible choices: Seurat, log

scale for the x-axis: Seurat (log(1+targetSum*X)) or log (log(X))

Default: "Seurat"

--targetSum factor when Seurat scale is used: (log(1+targetSum*X)) (default is 10^4, use 0 for the median of nRNA_Counts)

Default: 10000

--nx Number of values in x to check how your evaluated pdf is compatible with the model.

Default: 100

--osampx Oversampling factor of x values when evaluating pdf of Poisson distribution.

Default: 10

--osampxpdf Oversampling factor of x values when evaluating pdf at each step of the MCMC.

Default: 5

--minScale Minimal value of the scale of gaussians (Default is 0.1 but cannot be smaller than max of twice the bin size of pdf evaluation and half the bin size).

Default: 0.1

--nnorm Number of gaussian to fit.

Default: 2

--nsampMCMC Number of samplings (iteractions) of mcmc.

Default: 100000

--nsampBurnMCMC Number of samplings (iteractions) in the burning phase of mcmc (Default is nsampMCMC / 4).

--nsplitBurnMCMC Number of steps in the burning phase of mcmc.

Default: 10

--T0BurnMCMC Initial temperature in the burning phase of mcmc (>1).

Default: 100.0

--seed Change seed for another output.

Default: 1

--minNeff Will redo the MCMC with 10 times more samples until Neff is greater that this value (Default is not set so will not rerun MCMC).

--force Force to redo the mcmc even if output exists.

Optional arguments to get plots and text outputs

--figure Ouput figure filename.

--title Title in figures.

--removeFirstSamples Number of samples to ignore before making the plots (default is nsampMCMC / 4).

--nsampInPlot Approximate number of samples to use in plots.

Default: 100000

--prettyBins Number of bins to use in plots (Default is nx).

Optional arguments to get logevidence

--logevidence Ouput file to put logevidence value.

--coviscale Scale factor to apply to covariance of parameters to get random parameters in logevidence evaluation.

Default: 1

--nis Size of sampling of random parameters in logevidence evaluation.

Default: 1000

1.2.3 combineMultipleModels_1d

Combine mcmc results from multiple models to get a mixture using logevidence to infer weights.

```
usage: combineMultipleModels_1d [-h] --input INPUT --geneColName GENECOLNAME
                                [--metadata1ColName METADATA1COLNAME]
                                [--metadata1Values METADATA1VALUES]
                                [--metadata2ColName METADATA2COLNAME]
                                [--metadata2Values METADATA2VALUES]
                                [--metadata3ColName METADATA3COLNAME]
                                [--metadata3Values METADATA3VALUES] --outputs
                                OUTPUTS [OUTPUTS ...] [--xmin XMIN]
                                [--xmax XMAX] [--xscale {Seurat,log}]
                                [--targetSum TARGETSUM] [--nx NX]
                                [--osampx OSAMPX] [--osampxpdf OSAMPXPDF]
                                [--minScale MINSCALE] [--seed SEED] --figure
```

(continues on next page)

(continued from previous page)

```
                                FIGURE [--title TITLE]
                                [--removeFirstSamples REMOVEFIRSTSAMPLES]
                                [--nsampInPlot NSAMPINPLOT]
                                [--prettyBins PRETTYBINS]
                                [--logevidences LOGEVIDENCES [LOGEVIDENCES ...]]
                                [--coviscale COVISCALE] [--nis NIS]
                                [--version]
```

Named Arguments

--version show program's version number and exit

Required arguments

--input Input table with one line per cell columns with raw counts and one column nCount_RNA with total number of UMI per cell optionally other meta data to filter.

--geneColName Name of the column with gene counts.

--outputs Ouput files basename (will be npz) with different results of mcmc to combine.

--figure Ouput figure basename.

Optional arguments used to run MCMC

--xmin Minimum value to consider in x axis.

Default: 0

--xmax Maximum value to consider in x axis.

Default: 2.5

--xscale Possible choices: Seurat, log

scale for the x-axis: Seurat (log(1+targetSum*X)) or log (log(X))

Default: "Seurat"

--targetSum factor when Seurat scale is used: (log(1+targetSum*X)) (default is 10^4, use 0 for the median of nRNA_Counts)

Default: 10000

--nx Number of values in x to check how your evaluated pdf is compatible with the model.

Default: 100

--osampx Oversampling factor of x values when evaluating pdf of Poisson distribution.

Default: 10

--osampxpdf Oversampling factor of x values when evaluating pdf at each step of the MCMC.

Default: 5

--minScale Minimal value of the scale of gaussians (Default is 0.1 but cannot be smaller than max of twice the bin size of pdf evaluation and half the bin size).

Default: 0.1

--seed Change seed for another output.

Default: 1

Optional arguments to select input data

--metadata1ColName Name of the column with metadata1 to filter.

--metadata1Values Comma separated values for metadata1.

--metadata2ColName Name of the column with metadata2 to filter.

--metadata2Values Comma separated values for metadata2.

--metadata3ColName Name of the column with metadata3 to filter.

--metadata3Values Comma separated values for metadata3.

Optional arguments to customize plots and text outputs

--title Title in figures.

--removeFirstSamples Number of samples to ignore before making the plots (default is nsampMCMC / 4).

--nsampInPlot Approximate number of samples to use in plots.

Default: 100000

--prettyBins Number of bins to use in plots (Default is nx).

Optional arguments to evaluate logevidence

--logevidences Ouput files of precalculated log evidence values.(if not provided will be calculated).

--coviscale Scale factor to apply to covariance of parameters to get random parameters in logevidence evaluation.

Default: 1

--nis Size of sampling of random parameters in logevidence evaluation.

Default: 1000

1.2.4 baredSC_2d

Run mcmc to get the pdf in 2D for 2 given genes using a normal distributions.

```
usage: baredSC_2d [-h] --input INPUT --geneXColName GENEXCOLNAME
                  --geneYColName GENEYCOLNAME
                  [--metadata1ColName METADATA1COLNAME]
                  [--metadata1Values METADATA1VALUES]
                  [--metadata2ColName METADATA2COLNAME]
                  [--metadata2Values METADATA2VALUES]
                  [--metadata3ColName METADATA3COLNAME]
                  [--metadata3Values METADATA3VALUES] [--xmin XMIN]
                  [--xmax XMAX] [--nx NX] [--osampx OSAMPX]
                  [--osampxpdf OSAMPXPDF] [--minScalex MINSCALEX]
                  [--ymin YMIN] [--ymax YMAX] [--ny NY] [--osampy OSAMPY]
                  [--osampypdf OSAMPYPDF] [--minScaley MINSCALEY]
                  [--scalePrior SCALEPRIOR] [--scale {Seurat,log}]
                  [--targetSum TARGETSUM] [--nnorm NNORM]
                  [--nsampMCMC NSAMPMCMC] [--nsampBurnMCMC NSAMPBURNMCMC]
                  [--nsplitBurnMCMC NSPLITBURNMCMC] [--T0BurnMCMC T0BURNMCMC]
                  [--seed SEED] [--minNeff MINNEFF] [--force] --output OUTPUT
                  [--figure FIGURE] [--title TITLE]
                  [--splity SPLITY [SPLITY ...]]
                  [--removeFirstSamples REMOVEFIRSTSAMPLES]
                  [--nsampInPlot NSAMPINPLOT] [--prettyBinsx PRETTYBINSX]
                  [--prettyBinsy PRETTYBINSY] [--log1pColorScale]
                  [--logevidence LOGEVIDENCE] [--coviscale COVISCALE]
                  [--nis NIS] [--version]
```

Named Arguments

--version show program's version number and exit

Required arguments

--input Input table with one line per cell columns with raw counts and one column nCount_RNA with total number of UMI per cell optionaly other meta data to filter.

--geneXColName Name of the column with gene counts for gene in x.

--geneYColName Name of the column with gene counts for gene in y.

--output Ouput file basename (will be npz) with results of mcmc.

Optional arguments to select input data

--metadata1ColName Name of the column with metadata1 to filter.

--metadata1Values Comma separated values for metadata1.

--metadata2ColName Name of the column with metadata2 to filter.

--metadata2Values Comma separated values for metadata2.

--metadata3ColName Name of the column with metadata3 to filter.

--metadata3Values Comma separated values for metadata3.

Optional arguments to run MCMC

--xmin Minimum value to consider in x axis.

Default: 0

--xmax Maximum value to consider in x axis.

Default: 2.5

--nx Number of values in x to check how your evaluated pdf is compatible with the model.

Default: 50

--osampx Oversampling factor of x values when evaluating pdf of Poisson distribution.

Default: 10

--osampxpdf Oversampling factor of x values when evaluating pdf at each step of the MCMC.

Default: 4

--minScalex Minimal value of the scale of gaussians on x (Default is 0.1 but cannot be smaller than max of twice the bin size of pdf evaluation and half the bin size on x axis).

Default: 0.1

--ymin Minimum value to consider in y axis.

Default: 0

--ymax Maximum value to consider in y axis.

Default: 2.5

--ny Number of values in y to check how your evaluated pdf is compatible with the model.

Default: 50

--osampy Oversampling factor of y values when evaluating pdf of Poisson distribution.

Default: 10

--osampypdf Oversampling factor of y values when evaluating pdf at each step of the MCMC.

Default: 4

--minScaley Minimal value of the scale of gaussians on yx (Default is 0.1 but cannot be smaller than max of twice the bin size of pdf evaluation and half the bin size on y axis).

Default: 0.1

--scalePrior Scale of the truncnorm used in the prior for the correlation.

Default: 0.3

--scale Possible choices: Seurat, log

scale for the x-axis and y-axis: Seurat (log(1+targetSum*X)) or log (log(X))

Default: "Seurat"

--targetSum factor when Seurat scale is used: (log(1+targetSum*X)) (default is 10^4, use 0 for the median of nRNA_Counts)

Default: 10000

--nnorm Number of gaussian 2D to fit.

Default: 1

--nsampMCMC Number of samplings (iteractions) of mcmc.

Default: 100000

--nsampBurnMCMC Number of samplings (iteractions) in the burning phase of mcmc (Default is nsampMCMC / 4).

--nsplitBurnMCMC Number of steps in the burning phase of mcmc.

Default: 10

--T0BurnMCMC Initial temperature in the burning phase of mcmc.

Default: 100.0

--seed Change seed for another output.

Default: 1

--minNeff Will redo the MCMC with 10 times more samples until Neff is greater that this value (Default is not set so will not rerun MCMC).

--force Force to redo the mcmc even if output exists.

Optional arguments to get plots and text outputs

--figure Ouput figure basename.

--title Title in figures.

--splity Threshold value to plot the density for genex for 2 categories in geney values.

--removeFirstSamples Number of samples to ignore before making the plots (default is nsampMCMC / 4).

--nsampInPlot Approximate number of samples to use in plots.

Default: 100000

--prettyBinsx Number of bins to use in x in plots (Default is nx).

--prettyBinsy Number of bins to use in y in plots (Default is ny).

--log1pColorScale Use log1p color scale instead of linear color scale.

Default: False

Optional arguments to get logevidence

--logevidence Ouput file to put logevidence value.

--coviscale Scale factor to appy to covariance of parameters to get random parameters in logevidence evaluation.

Default: 1

--nis Size of sampling of random parameters in logevidence evaluation.

Default: 1000

1.2.5 combineMultipleModels_2d

Combine mcmc 2D results from multiple models to get a mixture using logevidence to infer weights.

```
usage: combineMultipleModels_2d [-h] --input INPUT --geneXColName GENEXCOLNAME
                                --geneYColName GENEYCOLNAME
                                [--metadata1ColName METADATA1COLNAME]
                                [--metadata1Values METADATA1VALUES]
                                [--metadata2ColName METADATA2COLNAME]
                                [--metadata2Values METADATA2VALUES]
                                [--metadata3ColName METADATA3COLNAME]
                                [--metadata3Values METADATA3VALUES] --outputs
                                OUTPUTS [OUTPUTS ...] [--xmin XMIN]
                                [--xmax XMAX] [--nx NX] [--osampx OSAMPX]
                                [--osampxpdf OSAMPXPDF]
                                [--minScalex MINSCALEX] [--ymin YMIN]
                                [--ymax YMAX] [--ny NY] [--osampy OSAMPY]
                                [--osampypdf OSAMPYPDF]
                                [--minScaley MINSCALEY] [--scale {Seurat,log}]
                                [--scalePrior SCALEPRIOR]
                                [--targetSum TARGETSUM] [--seed SEED] --figure
                                FIGURE [--title TITLE]
                                [--splity SPLITY [SPLITY ...]]
                                [--removeFirstSamples REMOVEFIRSTSAMPLES]
                                [--nsampInPlot NSAMPINPLOT]
                                [--prettyBins PRETTYBINS]
                                [--prettyBinsx PRETTYBINSX]
                                [--prettyBinsy PRETTYBINSY]
                                [--log1pColorScale] [--getPVal]
                                [--logevidences LOGEVIDENCES [LOGEVIDENCES ...]]
                                [--coviscale COVISCALE] [--nis NIS]
                                [--version]
```

Named Arguments

--version show program's version number and exit

Required arguments

--input Input table with one line per cell columns with raw counts and one column nCount_RNA with total number of UMI per cell optionally other meta data to filter.

--geneXColName Name of the column with gene counts for gene in x.

--geneYColName Name of the column with gene counts for gene in y.

--outputs Ouput files basename (will be npz) with different results of mcmc to combine.

--figure Ouput figure basename.

Optional arguments used to run MCMC

--xmin Minimum value to consider in x axis.

Default: 0

--xmax Maximum value to consider in x axis.

Default: 2.5

--nx Number of values in x to check how your evaluated pdf is compatible with the model.

Default: 50

--osampx Oversampling factor of x values when evaluating pdf of Poisson distribution.

Default: 10

--osampxpdf Oversampling factor of x values when evaluating pdf at each step of the MCMC.

Default: 4

--minScalex Minimal value of the scale of gaussians on x (Default is 0.1 but cannot be smaller than max of twice the bin size of pdf evaluation and half the bin size on x axis).

Default: 0.1

--ymin Minimum value to consider in y axis.

Default: 0

--ymax Maximum value to consider in y axis.

Default: 2.5

--ny Number of values in y to check how your evaluated pdf is compatible with the model.

Default: 50

--osampy Oversampling factor of y values when evaluating pdf of Poisson distribution.

Default: 10

--osampypdf Oversampling factor of y values when evaluating pdf at each step of the MCMC.

Default: 4

--minScaley Minimal value of the scale of gaussians on yx (Default is 0.1 but cannot be smaller than max of twice the bin size of pdf evaluation and half the bin size on y axis).

Default: 0.1

--scale Possible choices: Seurat, log

scale for the x-axis and y-axis: Seurat (log(1+targetSum*X)) or log (log(X))

Default: "Seurat"

--scalePrior Scale of the truncnorm used in the prior for the correlation.

Default: 0.3

--targetSum factor when Seurat scale is used: (log(1+targetSum*X)) (default is 10^4, use 0 for the median of nRNA_Counts)

Default: 10000

--seed Change seed for another output.

Default: 1

Optional arguments to select input data

--metadata1ColName Name of the column with metadata1 to filter.

--metadata1Values Comma separated values for metadata1.

--metadata2ColName Name of the column with metadata2 to filter.

--metadata2Values Comma separated values for metadata2.

--metadata3ColName Name of the column with metadata3 to filter.

--metadata3Values Comma separated values for metadata3.

Optional arguments to customize plots and text outputs

--title Title in figures.

--splity Threshold value to plot the density for genex for 2 categories in geney values.

--removeFirstSamples Number of samples to ignore before making the plots (default is nsampMCMC / 4).

--nsampInPlot Approximate number of samples to use in plots.

Default: 100000

--prettyBins Number of bins to use in plots (Default is nx).

--prettyBinsx Number of bins to use in x in plots (Default is nx).

--prettyBinsy Number of bins to use in y in plots (Default is ny).

--log1pColorScale Use log1p color scale instead of linear color scale.

Default: False

--getPVal Use less samples to get an estimation of the p-value.

Default: False

Optional arguments to evaluate logevidence

--logevidences Ouput files of precalculated log evidence values.(if not provided will be calculated).

--coviscale Scale factor to apply to covariance of parameters to get random parameters in logevidence evaluation.

Default: 1

--nis Size of sampling of random parameters in logevidence evaluation.

Default: 1000

1.3 Outputs

        * *name_splitX_renorm.extension*
        * *name_splitX.txt*
    - *Evidence*

We will describe here each of the output file.

1.3.1 MCMC output

The only output by defaut is a numpy compressed `.npz` file. This output contains the result of the MCMC:

- samples: the value of the parameter at each step of the MCMC
- diagnostics: a dictionary with the diagnostics at each step of the MCMC, among them:
  - logprob: the log probability at each step of the MCMC
  - mu: the final estimate of the mean of each parameter
  - cov: the final estimate of the covariance matrix of parameters
- some of the input values

When the tool is run while the output exists it will use it instead of rerunning it which is useful to get more plots.

1.3.2 Plots and txt outputs

When `--figure name.extension` is given then some QC and results are given.

QC

name_convergence.extension

This plot shows the autocorrelation between samples for all parameters (the solid line shows the median, the shaded area shows the min and max). If the MCMC converged, you should see a value of ACF close to 0 since a small value of T.

name_neff.txt

From the autocorrelation displayed above, we can evaluate the number of independent samples, also called effective sample size. The value is printed and stored in this text file.

name_p.extension

This plot shows the value of each parameter and the log probability (y axis, one panel per parameter) for all samples (x-axis). When the MCMC did not converged, it can be helpful to see if it can be explained by the fact that the first samples considered where not around the final solution. In this case, it can be useful to rerun the plots using an increase value of `--removeFirstSamples` (by default it is 1/4 of the number of samples), or increase the number of samples.

name_corner.extension

This plot shows the distribution of value of each parameter in relationship the one with the other. It can help to see which parameters are correlated. Also, when the MCMC did not converged, it can help to identify if 2 or more solutions were explored.

Results

name.extension

This is the figure with the results.

- When the 1d version is used, it displays the mean pdf in solid red line, the median in black dashed lines (/!the integral of the median is not equal to 1) with the confidence interval of 1 sigma (68%), 2 sigma (95%) and 3 sigma (99.7%) as well as in green, the kernel density estimate of the input values, the detected expression (`log(1 + 10^4 * raw / total UMI)`).
- When the 2d version is used, it displays the pdf as a heatmap as well as a projection on the x and y axis. On the projection, the confidence interval 68% is indicated as a shaded area as well as the mean with a solid red line and the median with a dashed black line. On the top right corner, the correlation is indicated with the confidence interval 68% as well as a confidence interval on the one-sided p-value (the probability that the correlation is the opposite sign of the mean, one sigma confidence interval).

name_individuals.extension

- When the 1d version is used, it displays the pdf of 100 samples.
- When the 2d version is used, it displays the projection of the pdf of 100 samples.

name_p.txt

This is a tabulated delimited table with the 16 percentile (low), median, 84 percentile (high) value of each parameter.

name_pdf.txt (1d only)

For each value of x, the 16 percentile (low), mean, 84 percentile (high) and median, is given in a tabulated delimited file.

name_with_posterior.extension (1d only)

Same as name.extension except that a new orange line is plotted showing the posterior density evaluated as the average of the posterior density of each cell.

name_posterior_per_individuals.extension (1d only)

Showing posterior density probability of 50 random cells.

name_posterior_per_cell.txt (1d only)

For each cell of the input, providing the posterior average and standard deviation of the density probability.

name_posterior_andco.extension (1d only)

Showing the mean pdf, the median pdf, the density from raw counts normalized, the average of the posterior density from all cells, the density and a histogram using only the average value of the posterior distribution of each cell and the posterior density approximating the pdf of each cell by a Gaussian using values in the "posterior_per_cell.txt" file.

name_median.extension (2d only)

Same as name.extension except that the median instead of the mean is used.

name_corr.txt (2d only)

The mean, median, 16 percentile, 84 percentile, p-value and error on the p-value for the correlation (see above).

name_pdf2d.txt (2d only)

The mean pdf and the x and y values stored in a tabulated delimited file in a matrix format. Different x values correspond to different columns while different y values correspond to different rows.

name_pdf2d_flat.txt (2d only)

The x, y, 16 percentile (low), mean, 84 percentile (high) and median of pdf in a tabulated delimited file.

Results when `--splity` is provided in 2d

When `--splity` is provided the pdf above and below this threshold on the y axis are summed up, resulting in 2 pdf along the x axis.

name_splitX.extension

This plot shows the 2 pdfs. The ratio between the area represent the ratio of cells above and below the threshold of the gene y. The pdf for cells below the threshold is in red (with the shaded area for the 68% confidence interval) and the pdf for cells above the threshold is in green. In black is the pdf of all cells projected on the x axis (sum of the 2).

name_splitX_renorm.extension

Same plot as above except that the pdf were renormalized so the area of each pdf is equal to 1. Also the median is added in dashed black lines.

name_splitX.txt

This is a tabulated delimited table with the x values, the 16 percentile (low), mean, 84 percentile (high) values of each pdf (below and above the threshold) before normalization.

1.3.3 Evidence

When `--logevidence` is set. The log evidence is calculated and stored in this file. This can be used to compare different models, here different number of gaussians.

1.4 Tutorial on simulated data

1.4.1 Run baredSC_1d with default parameters

We will describe here an example step by step on simulated data where we will use default parameters and carefully check the QC.

Inputs

We took total UMI counts from a real dataset of NIH3T3. We generated a example where 2 genes have the same distribution (2 gaussians, one of mean 0.375, scale 0.125 and another one of mean 1 and scale 0.1). Half of cells goes in each gaussian. The gene is called "0.5_0_0_0.5_x".

Run

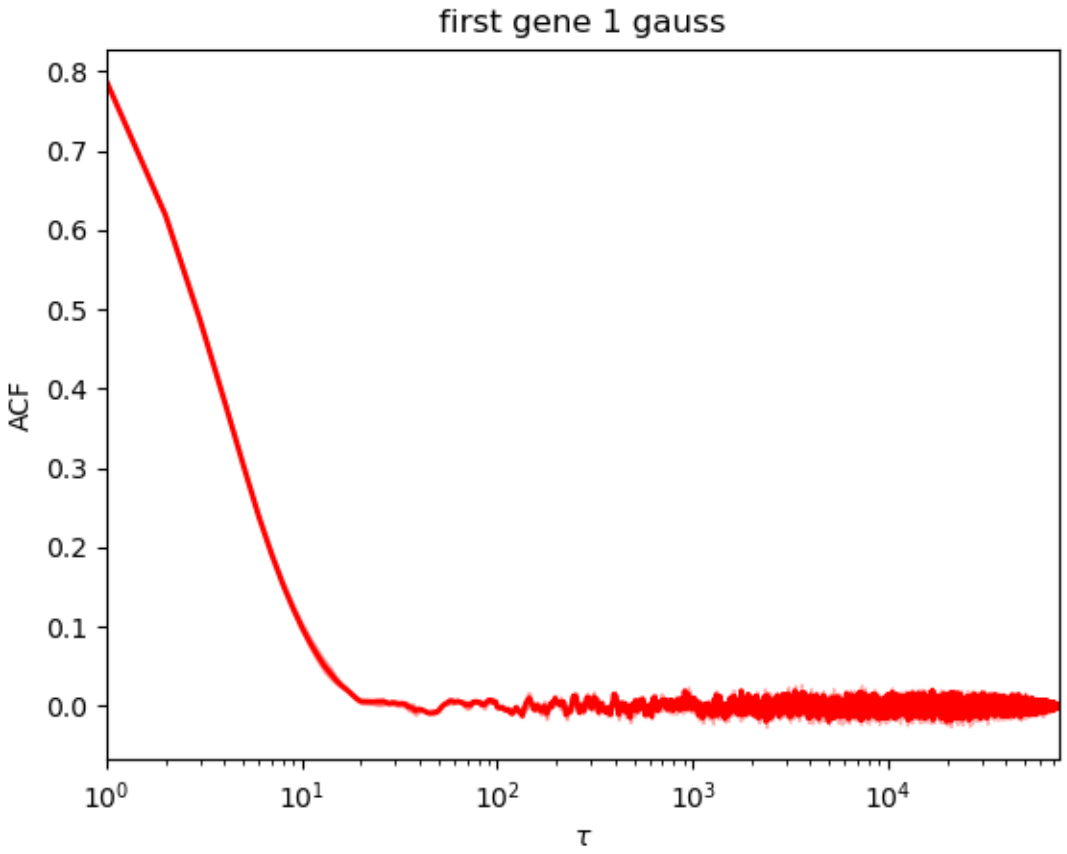
Run 1 gaussian

Let say we don't know the number of gaussian, we try one. We keep the default parameters and we set `--figure` to get visual outputs:

```
$ baredSC_1d \
    --input example/nih3t3_generated_2d_2.txt \
    --geneColName 0.5_0_0_0.5_x \
    --output example/first_example_1d_1gauss \
    --nnorm 1 \
    --figure example/first_example_1d_1gauss.png \
    --title "first gene 1 gauss" \
    --logevidence example/first_example_1d_1gauss_logevid.txt
```

Check QC

We first check the convergence:

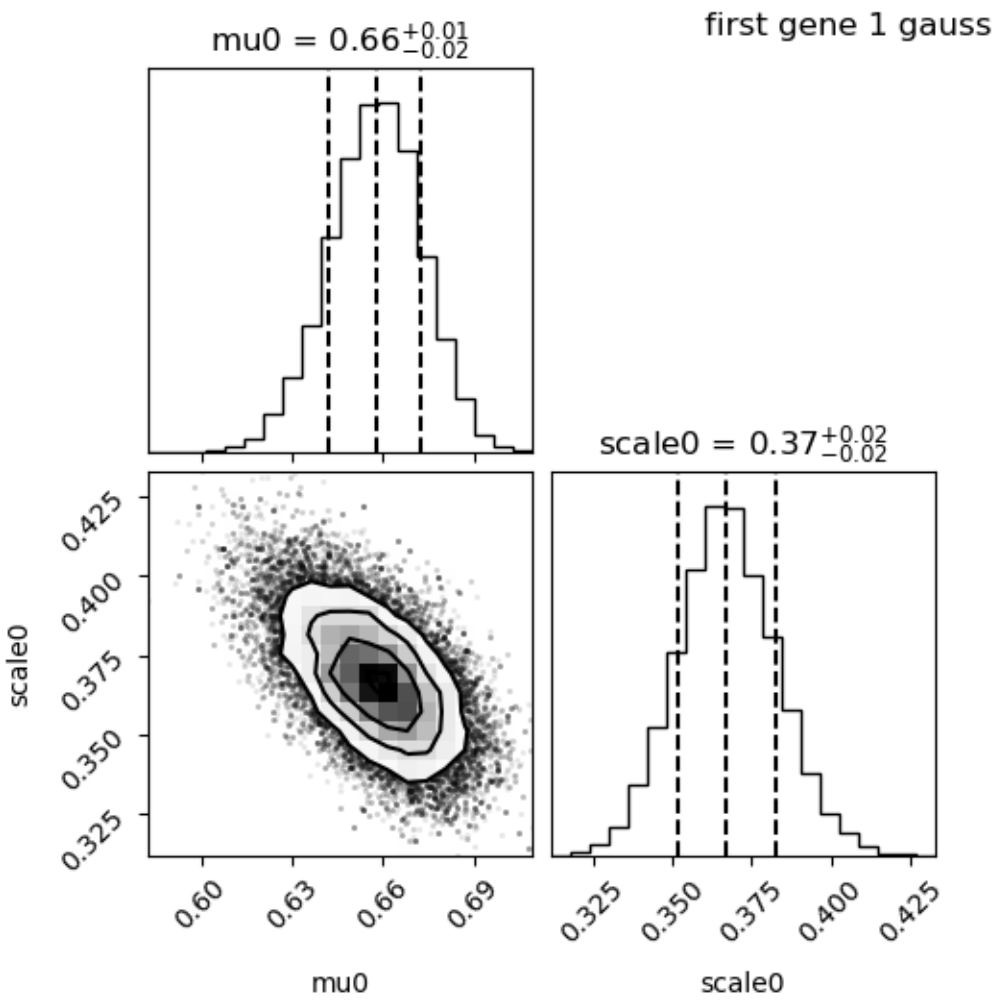


This plot show the autocorrelation as a function of number of samples. The earlier the curves goes close to 0, the more it convereged.

Here, this is perfect.

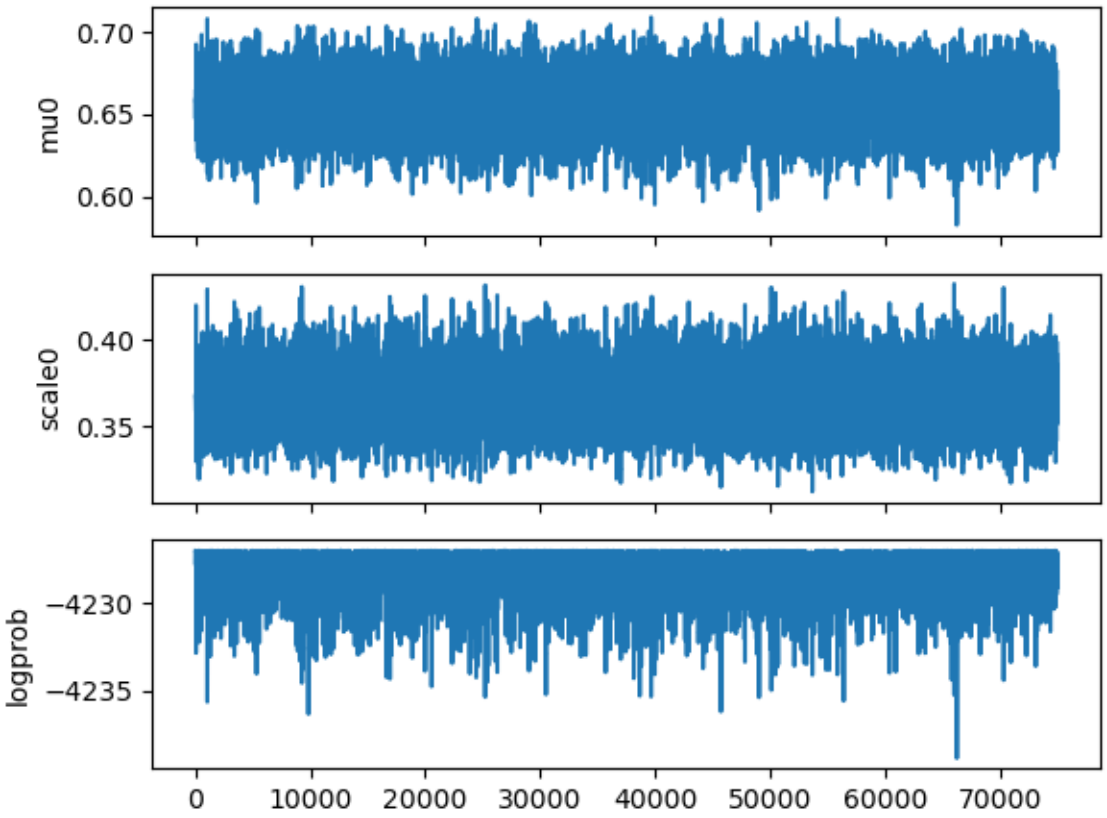
As printed during the run (or reading the file *neff.txt), the Neff is around 8000 which is enough to estimate the confidence interval.

We have a look at the corner plot:



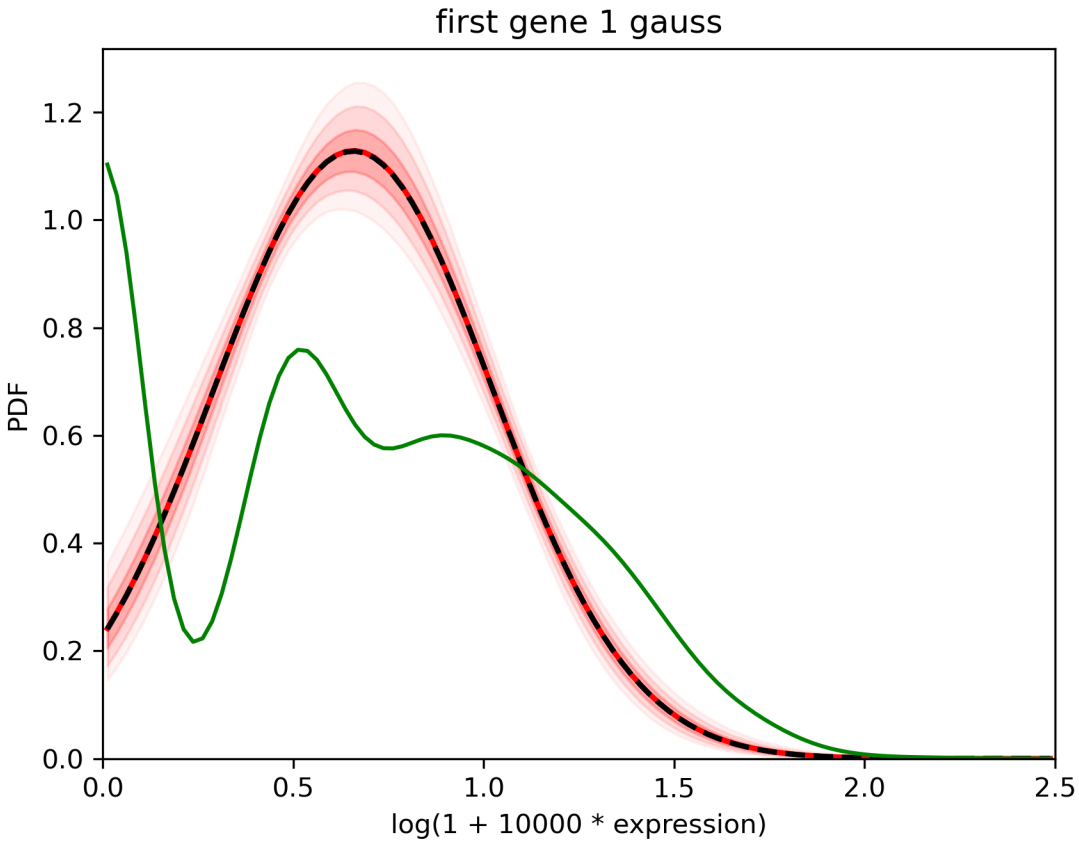
We see that the distribution of each parameter is close to gaussian, this is perfect.

We have a look at the parameter plot:



It nicely shows that the 2 parameters (mu0 and scale0) oscilate around the mean position while the log probability oscilate with a maximum value.

Look at the results



The pdf is well constrained (the shaded areas indicating the 68%, 95% and 99% interval are thin). The mean in red is really close to the median in dashed black line. However, using the green curve which is the density of the detected expression, we suspect that there are 2 gaussians.

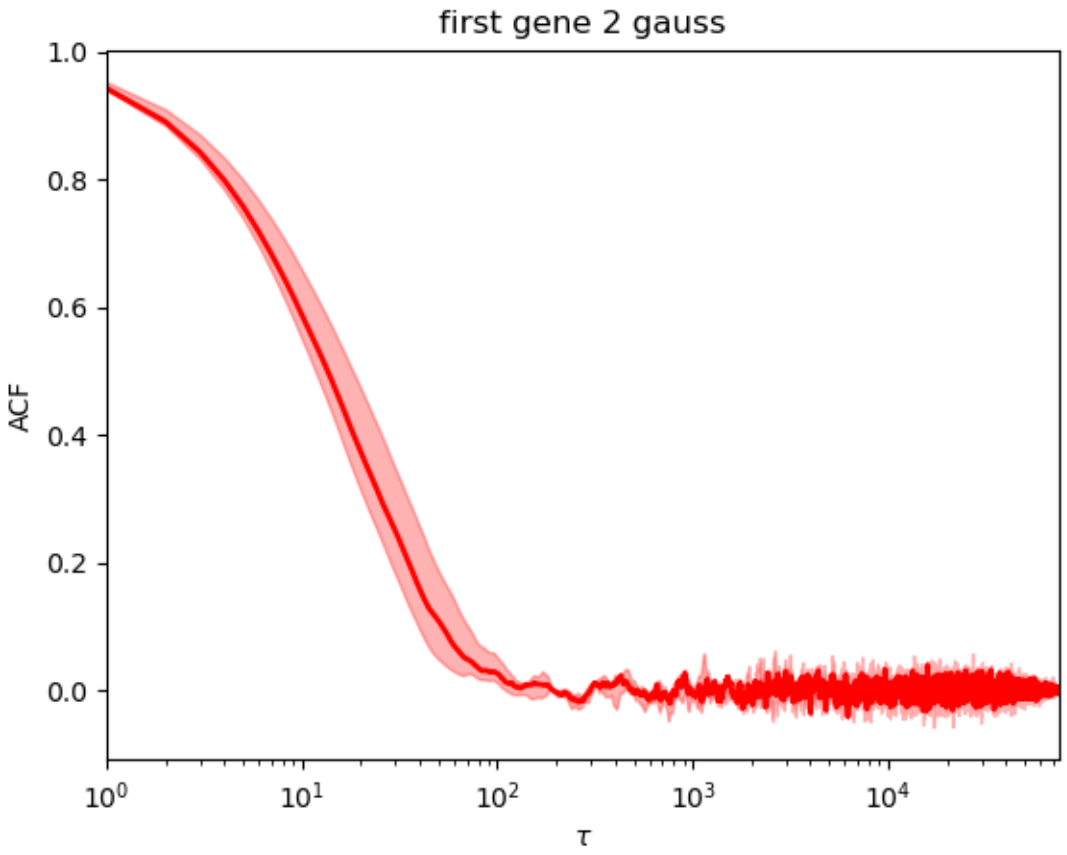
Run 2 gaussians

Now let's try 2 gaussians

```
$ baredSC_1d \
    --input example/nih3t3_generated_2d_2.txt \
    --geneColName 0.5_0_0_0.5_x \
    --output example/first_example_1d_2gauss \
    --nnorm 2 \
    --figure example/first_example_1d_2gauss.png \
    --title "first gene 2 gauss" \
    --logevidence example/first_example_1d_2gauss_logevid.txt
```

Check QC

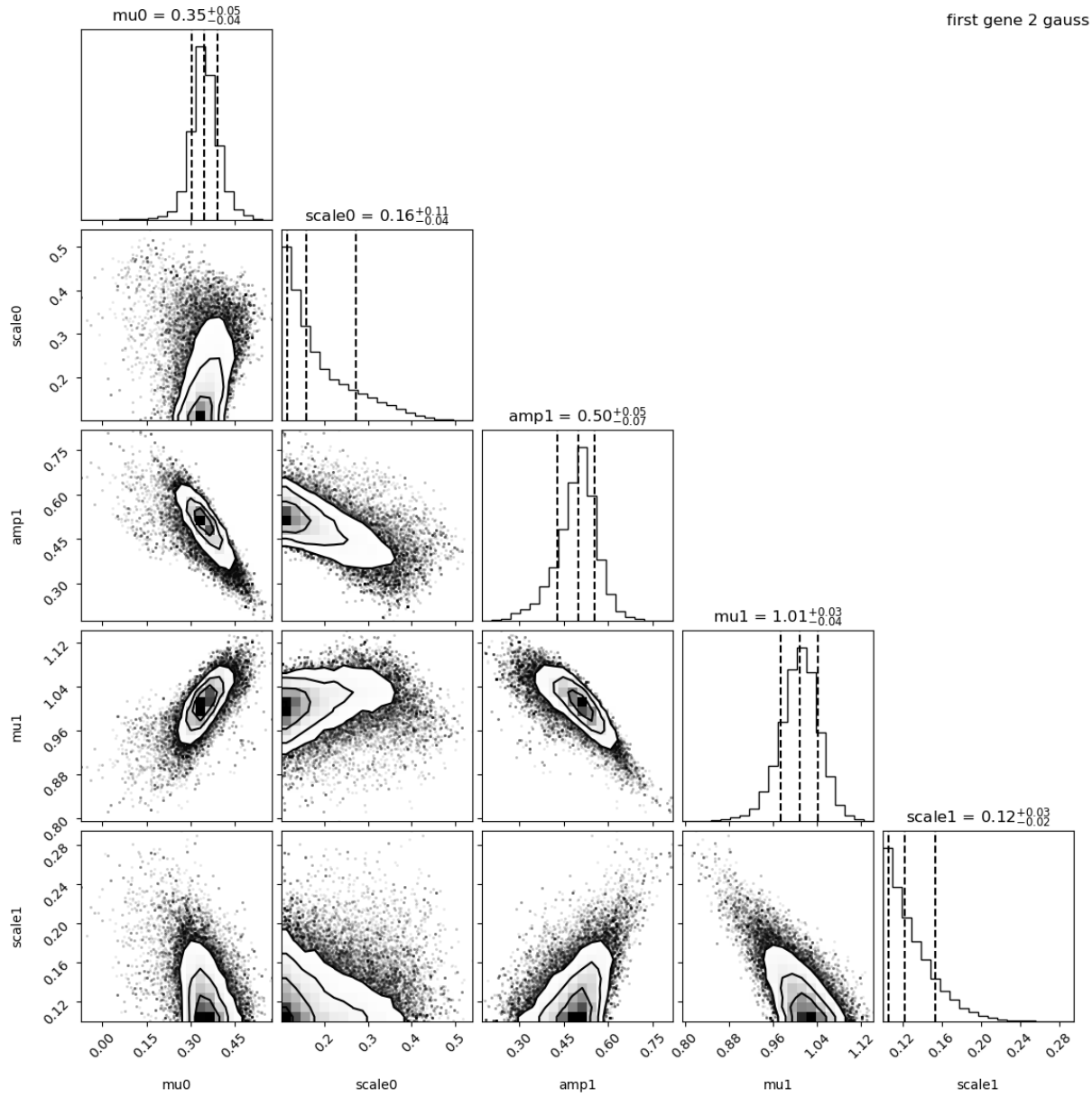
We first check the convergence:



This is perfect.

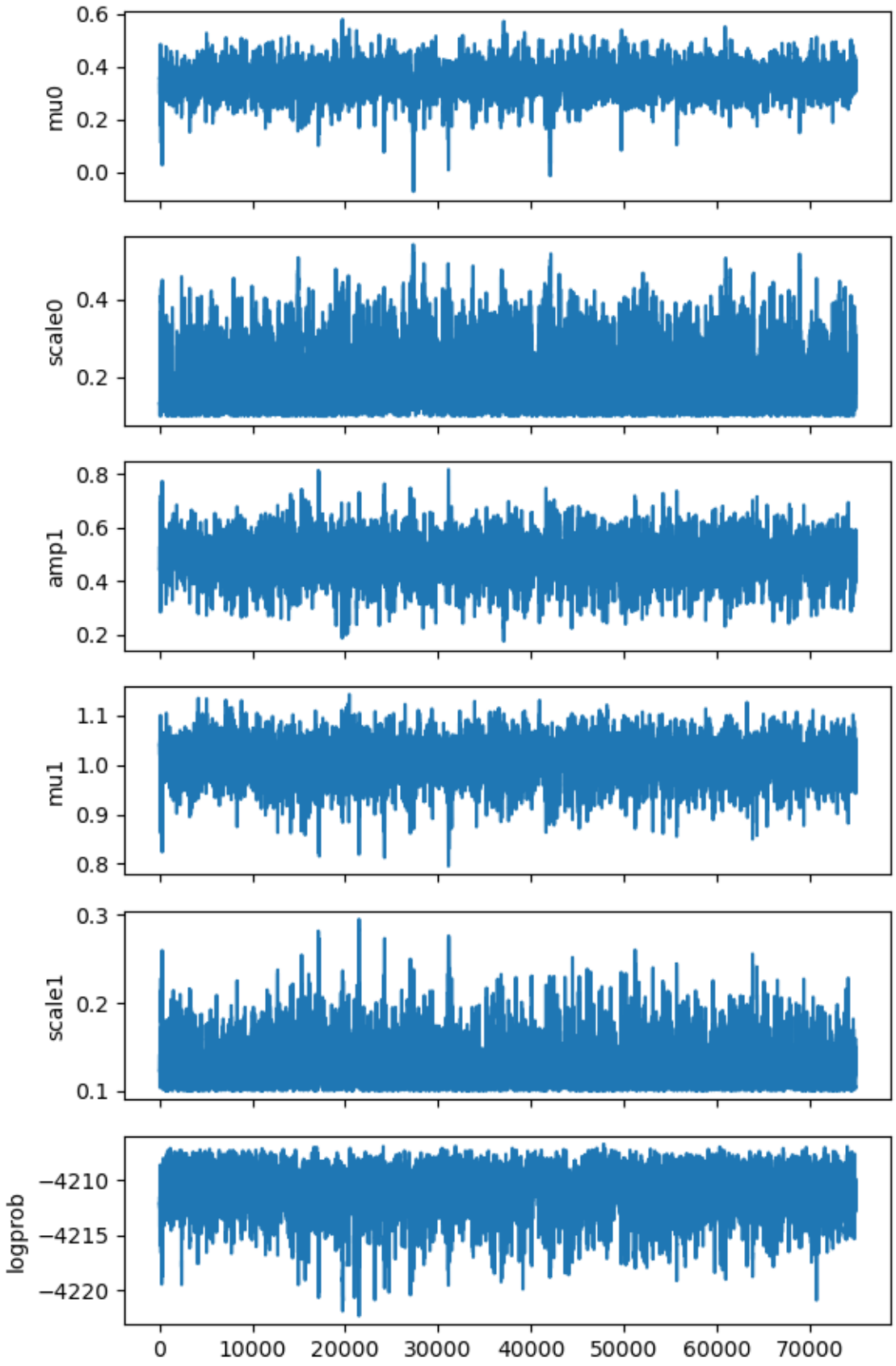
As printed during the run (or reading the file *neff.txt), the Neff is around 1300, perfect.

We have a look at the corner plot:



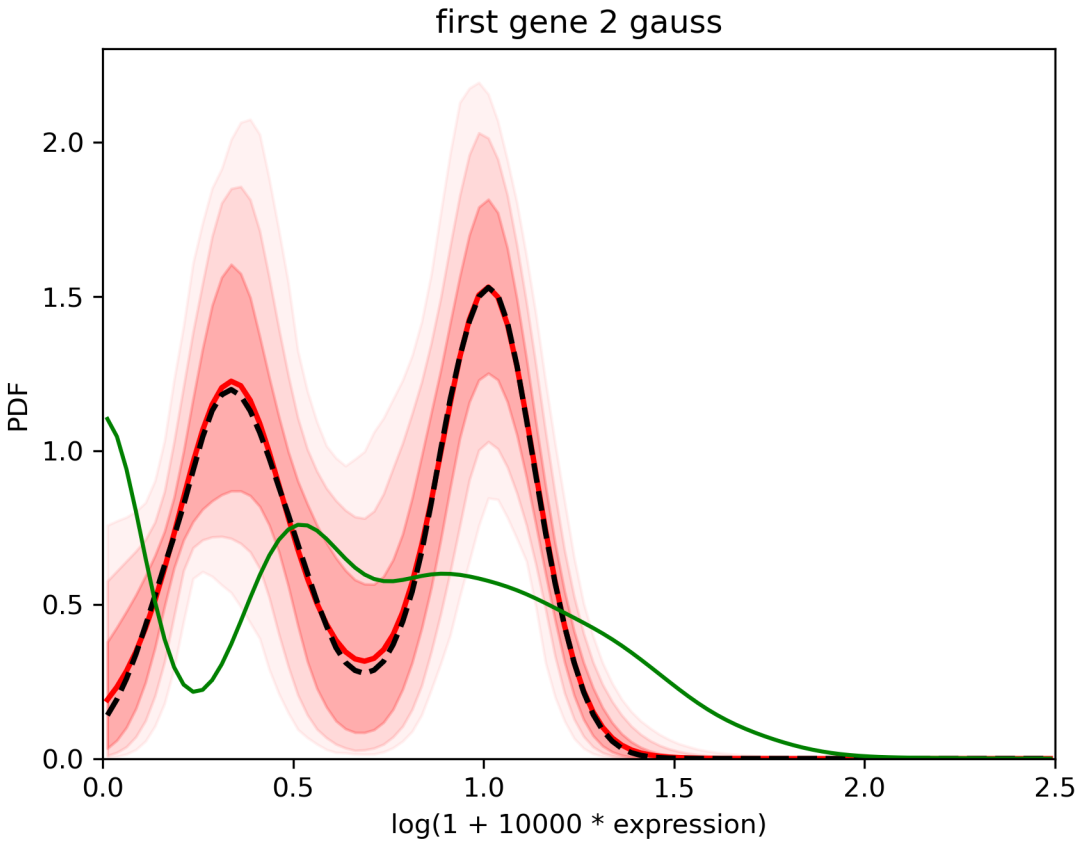
The means and amplitude are like a gaussian. The scale distribution is asymetric because by default, the minimum scale is set to 0.1 which is close to our values here. Some parameters are correlated: the mean of the first Gaussian with the mean of the second Gaussian. Some are anti-correlated: the mean of the second Gaussian with its amplitude. But this is not problematic, just an information we can get from this plot.

We have a look at the parameter plot:


mu0
0.6
0.4
0.2
0.0
scale0
0.4
0.2
amp1
0.8
0.6
0.4
0.2
mu1
1.1
1.0
0.9
0.8
scale1
0.3
0.2
0.1
logprob
−4210
−4215
−4220
0
10000
20000
30000
40000
50000
60000
70000

It nicely shows that the 5 parameters oscilate around the mean position and the log probability is quite constant.

Look at the results



The confidence interval is larger than in the first case but still good.

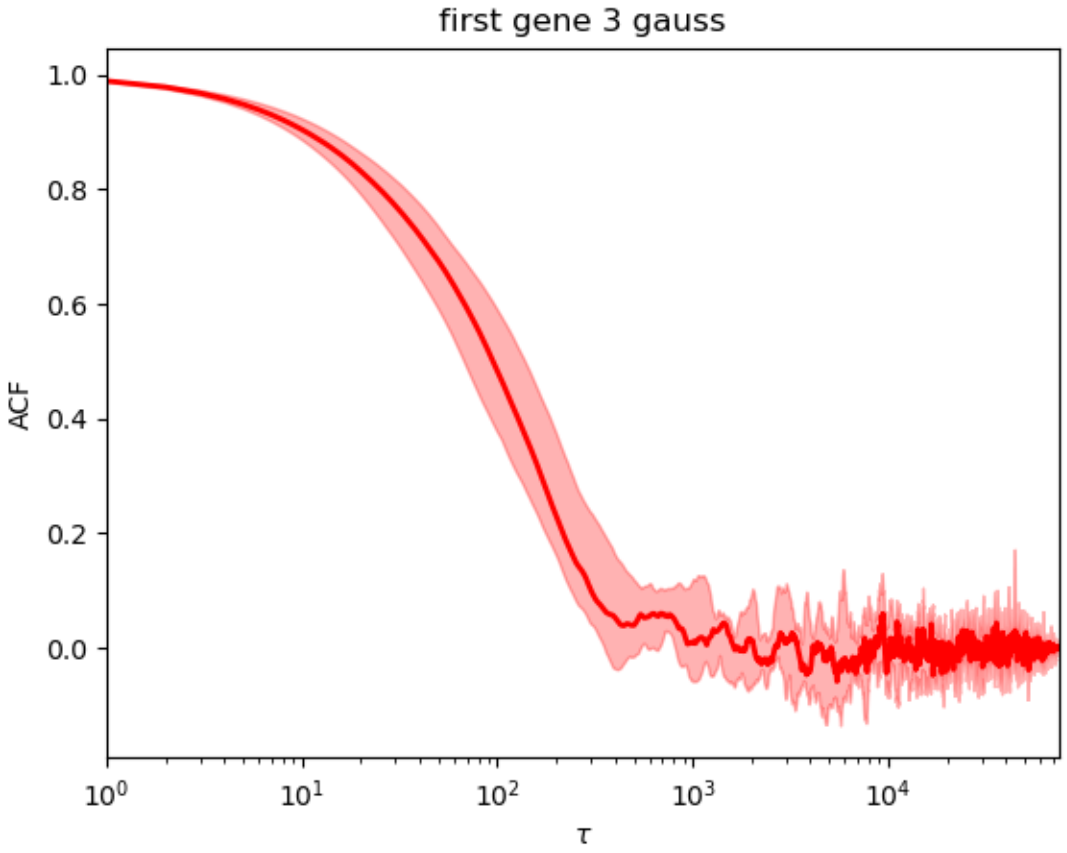
Run 3 gaussians

Now let's try 3 gaussians

```
$ baredSC_1d \
    --input example/nih3t3_generated_2d_2.txt \
    --geneColName 0.5_0_0_0.5_x \
    --output example/first_example_1d_3gauss \
    --nnorm 3 \
    --figure example/first_example_1d_3gauss.png \
    --title "first gene 3 gauss" \
    --logevidence example/first_example_1d_3gauss_logevid.txt
```

Check QC

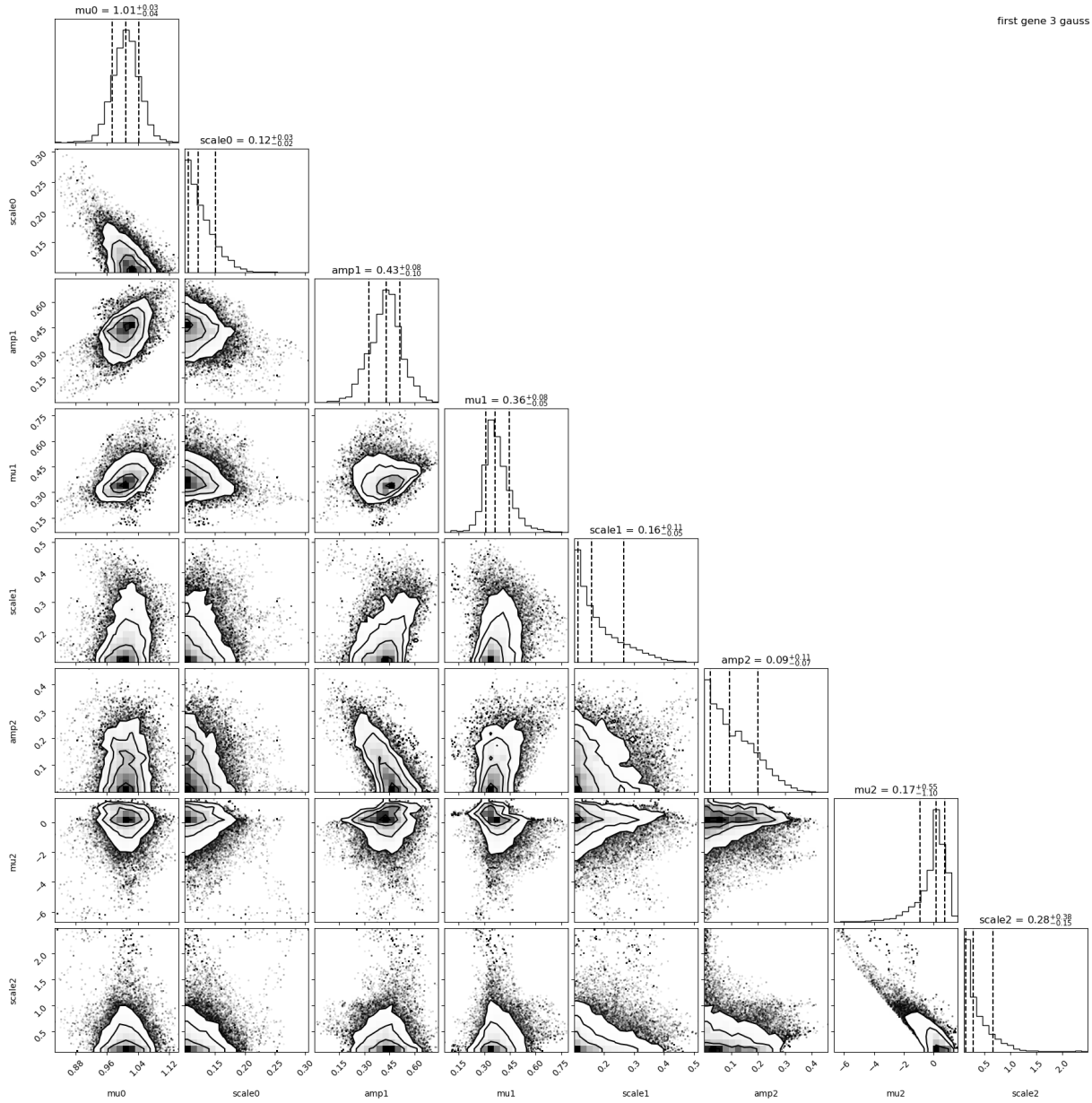
We first check the convergence:



It is much worse than the first ones. The auto-correlation decreases later and does not stay a flat line at 0 but oscillate.

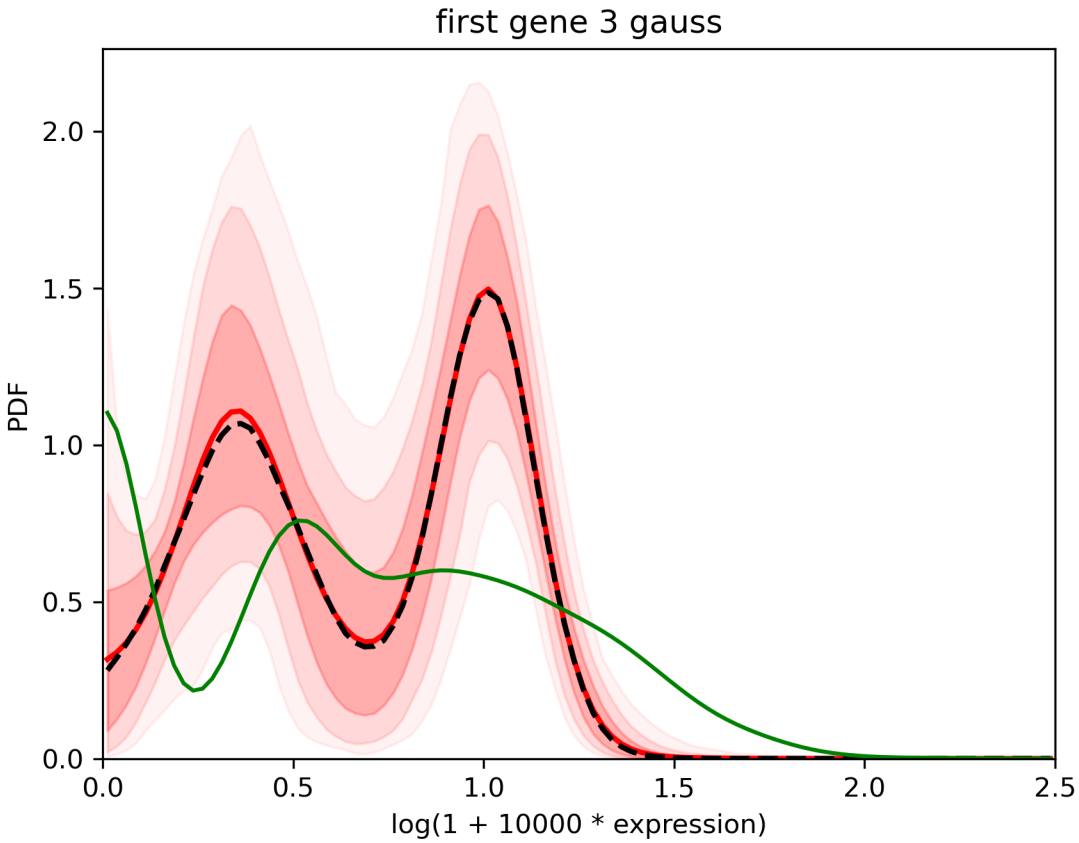
As printed during the run, the Neff is around 191. This is better to get more indenpendent samples. We can rerun with another value of the seed but it is safer to rerun with increased number of samples.

We still have a look at the corner plot:



The first two Gaussians are close to what was expected. The third Gaussian is a Gaussian with a reduced mean (0.17 in average). We see that this last Gaussian is not very well constrained (large error bar on each of its parameters).

Look at the results

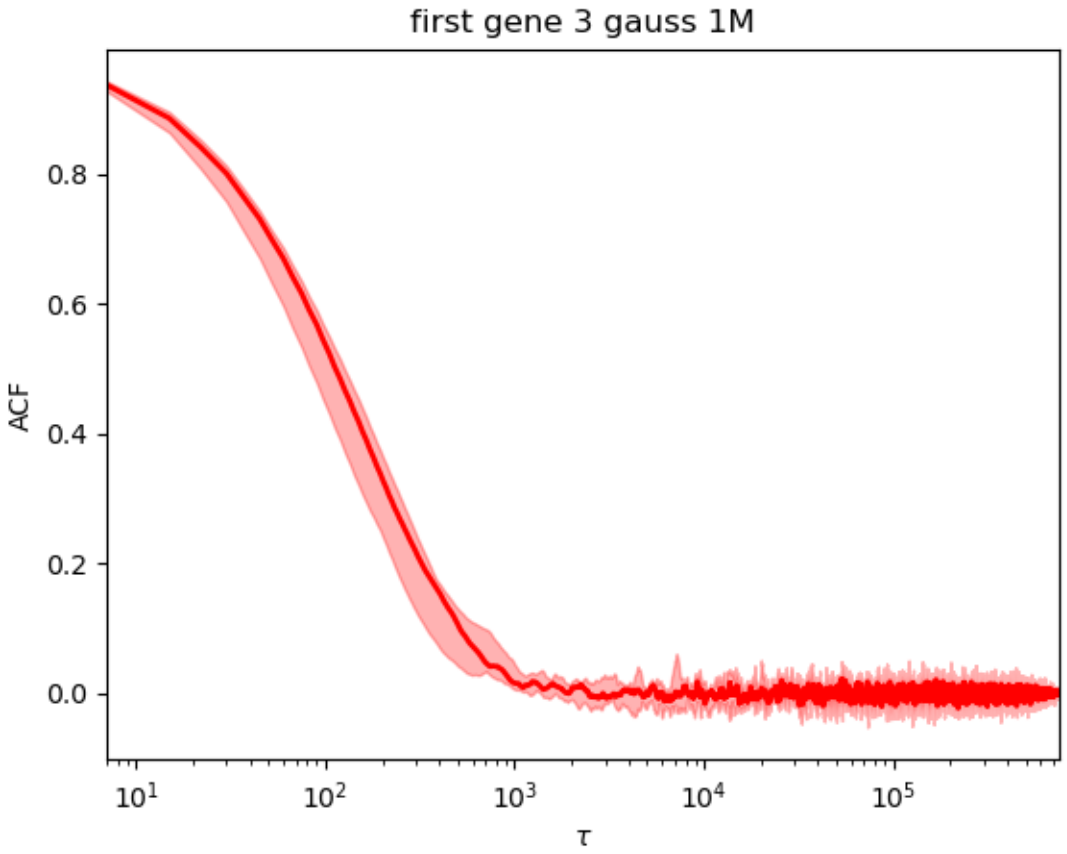


The results are very close to the one with 2 Gaussians.

Rerun 3 gauss with more samples

```
$ baredSC_1d \
    --input example/nih3t3_generated_2d_2.txt \
    --geneColName 0.5_0_0_0.5_x \
    --output example/first_example_1d_3gauss_1M \
    --nnorm 3 --nsampMCMC 1000000 \
    --figure example/first_example_1d_3gauss_1M.png \
    --title "first gene 3 gauss 1M" \
    --logevidence example/first_example_1d_3gauss_1M_logevid.txt
```

It converged:



Automatic rerun when Neff is too small

While some models converge even with a small number of samples, some other needs a lot of sample to reach acceptable covergence. A way to automatically rerun the MCMC when the effectif number of samples is too low is to use the `--minNeff`. The MCMC will be rerun with 10 times more sample until it reaches the value. This can potentially take forever as some model may never converge. But can be useful in other cases. Even with this option, we highly encourage the users to manually check the QC.

Compare models

In order to compare models, we will use the values of logevidence.

model	log evidence
1gauss	-4233.7
2gauss	-4221.8
3gauss	-4223.0

We can see that the model with the highest log evidence is the model with 2 gaussians. However, we see that the model with 3 gaussians is very close. When you compare models, what is important is the difference between the log evidence, not its absolute value.

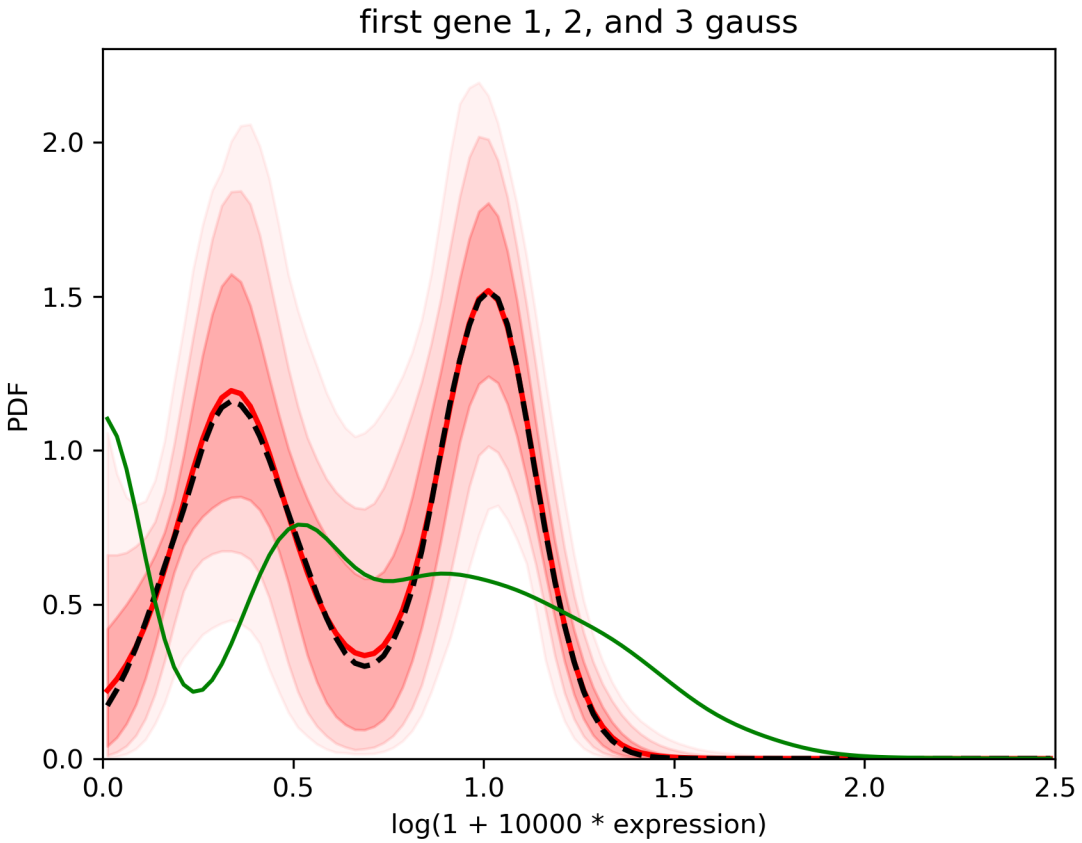
We can either choose the best model or decide to combine them:

Combine models

Another way to use these models is to use samples from all models but using the log evidence to put weight on the number of sample to use from each model.

```
$ combineMultipleModels_1d \
    --input example/nih3t3_generated_2d_2.txt \
    --geneColName 0.5_0_0_0.5_x \
    --outputs example/first_example_1d_1gauss \
    example/first_example_1d_2gauss \
    example/first_example_1d_3gauss_1M \
    --figure example/first_example_1d_1-3gauss.png \
    --title "first gene 1, 2, and 3 gauss"
```

In the standard output you will see that it only integrates samples from the 2gauss and 3gauss. Here is the result:



1.4.2 Run baredSC_2d with default parameters

Inputs

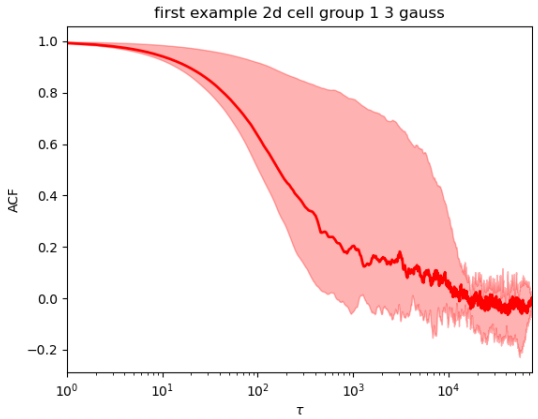
We took total UMI counts from a real dataset of NIH3T3. We generated a example where 2 genes have the same distribution (2 gaussians, one of mean 0.375, scale 0.125 and another one of mean 1 and scale 0.1). For each gene, half of cells goes in each gaussian. The genes are called "0.5_0_0_0.5_x" and "0.5_0_0_0.5_y".

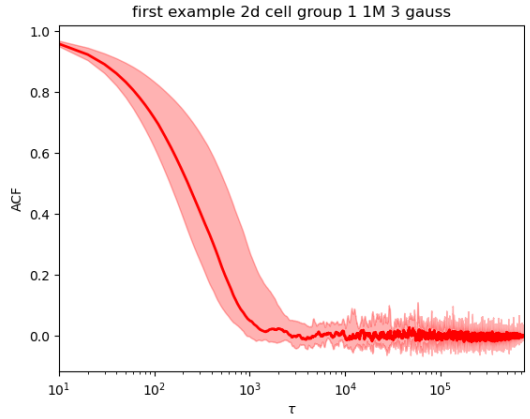
2d

As for the 1d, you need to run it with different number of gaussian 2d to find the best model or to mix them. The 2d tool is much slower than the 1d. The time depends on the `--nx`, `--ny`, `--osampxpdf`, `--osampypdf` parameters and the number of cells. To make the example quicker we will run only on the 300 random cells (group1).

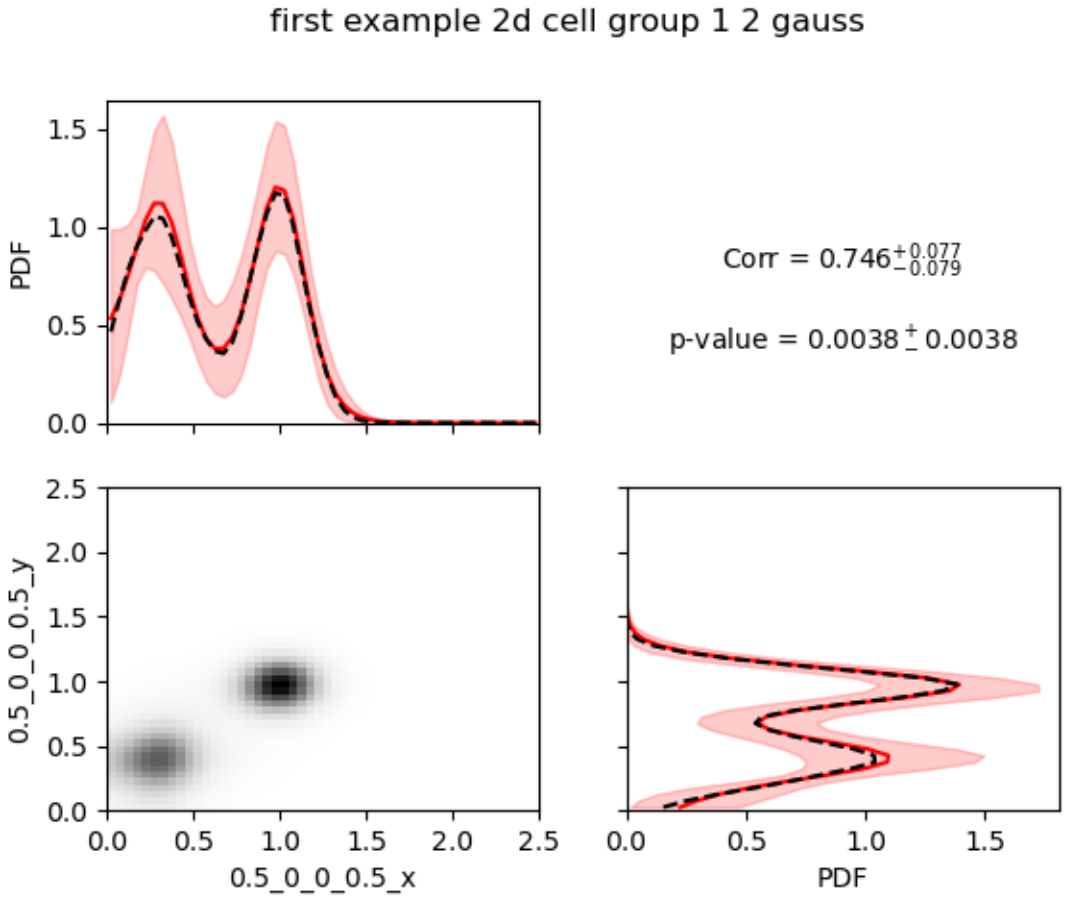
```
$ for nnorm in 1 2 3; do
    baredSC_2d \
      --input example/nih3t3_generated_2d_2.txt \
      --geneXColName 0.5_0_0_0.5_x \
      --geneYColName 0.5_0_0_0.5_y \
      --metadata1ColName group \
      --metadata1Values group1 \
      --output example/first_example_2d_cellgroup1_${nnorm}gauss \
      --nnorm ${nnorm} \
      --figure example/first_example_2d_cellgroup1_${nnorm}gauss.png \
      --title "first example 2d cell group 1 ${nnorm} gauss" \
      --logevidence example/first_example_cellgroup1_2d_${nnorm}gauss_logevid.txt
  done
```

The QC are done the same way as the 1d. The models with 1 and 2 gaussians are converging. The model with 3 gaussian is not converging (picture below left). When we rerun it with 1 million samples, it now converges (picture below right).





The best model (using the log evidence) is the 2 gaussians model.



We see a very high correlation highly significant. Here, we would like to warn the users that the correlation calculated here is a Pearson correlation, so it reflects how much the data are close to a line with positive or negative slope.

In order to appreciate the confidence interval it can be useful to split the 2d pdf in 2 parts: one above a threshold for y and one below the same threshold. This is for this purpose that we can use `--splity`. For the demo we will try different values:

```
$ baredSC_2d \
```

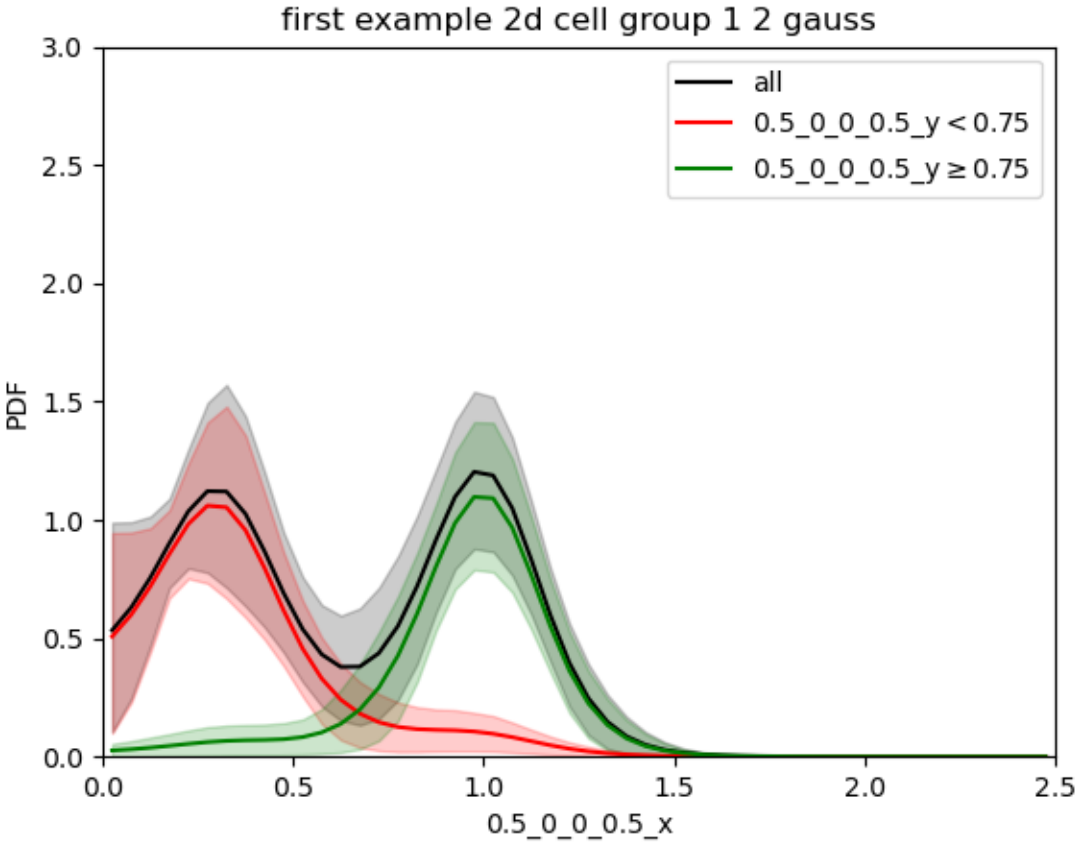
(continues on next page)

(continued from previous page)

```
    --input example/nih3t3_generated_2d_2.txt \
    --geneXColName 0.5_0_0_0.5_x \
    --geneYColName 0.5_0_0_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --output example/first_example_2d_cellgroup1_2gauss \
    --nnorm 2 \
    --figure example/first_example_2d_cellgroup1_2gauss.png \
    --title "first example 2d cell group 1 2 gauss" \
    --splity 0.35 0.75
```

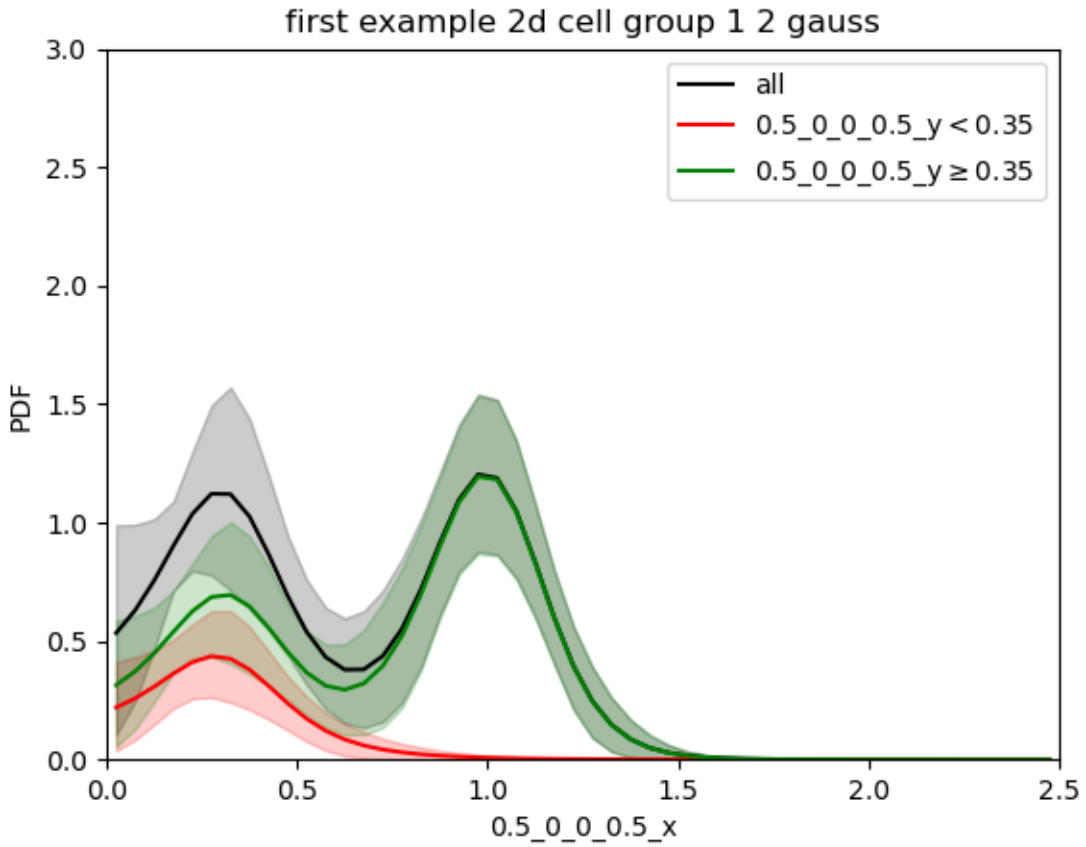
As the MCMC was run previously, it will use the `.npz` output to generate the figures, thus this operation is really quick.

When we split at 0.75 (between the 2 gaussian):

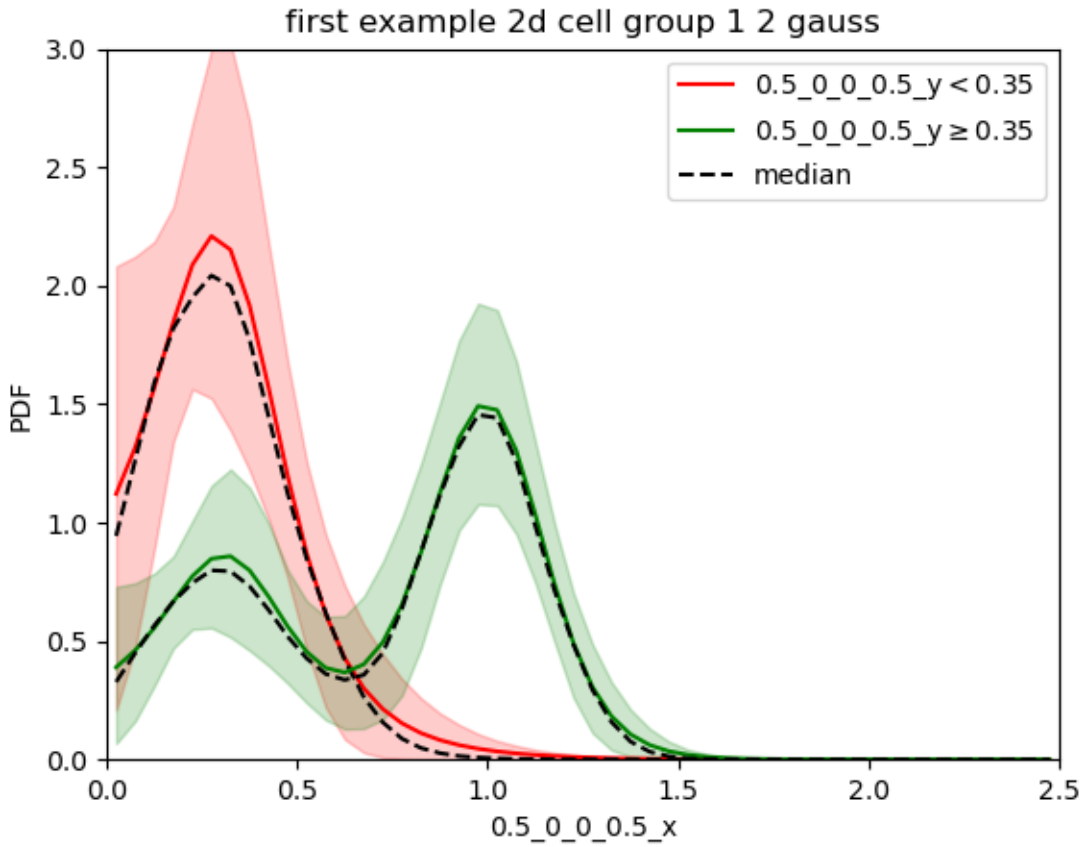


We find each of the 2 gaussians in 1d and the confidence interval is quite small.

When we split in the low gaussian (0.35):



We see that the green curve is made of 2 gaussian. The sum of both the green and red curves is the black one. This can make the comparison difficult. So the output `renorm.extension` is sometimes better.



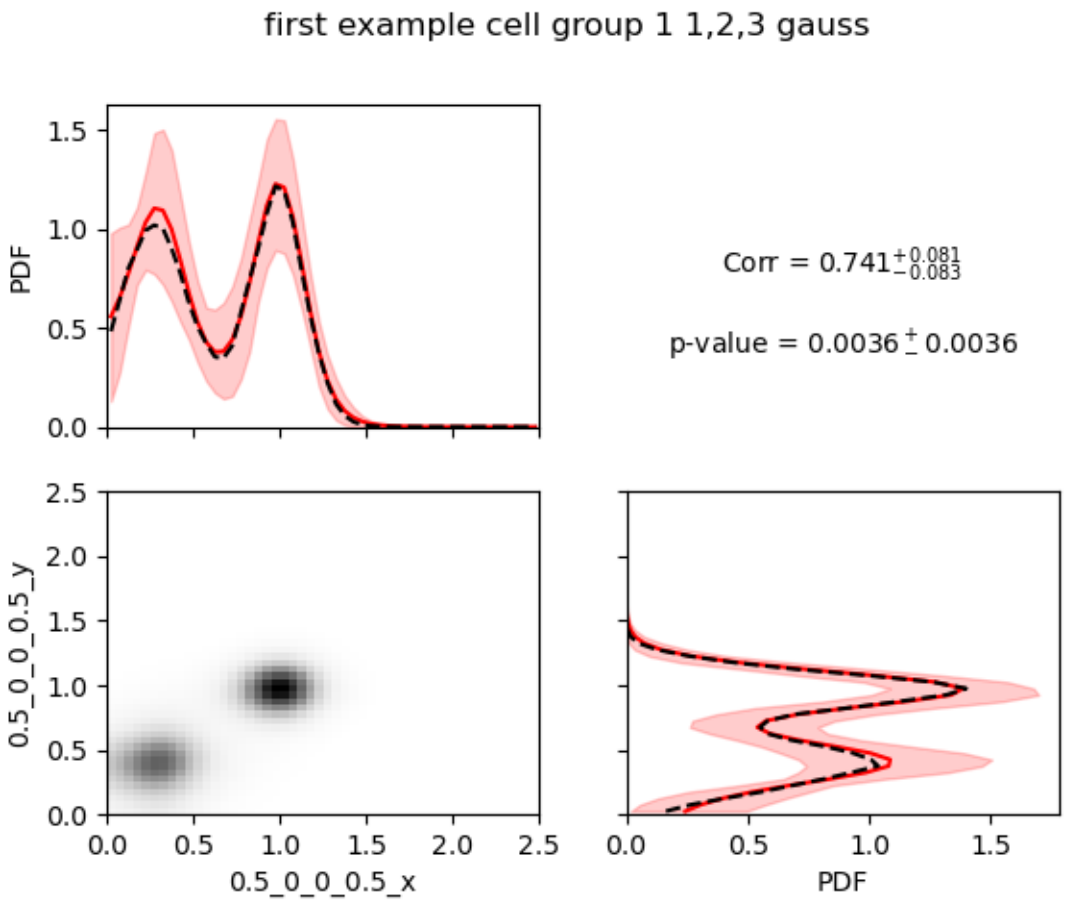
Now we clearly see that in the cells with low expression of gene y all cells are low for gene x while for cells with relatively high expression of gene y, gene x is bimodal with a greater proportion in the second gaussian.

Similarly to the 1d, the option `--minNeff` is also implemented.

You can combine multiple models with `combineMultipleModels_2d`. By default, no p-value will be evaluated for the correlation but you can use less samples to get a p-value with `--getPVal`.

```
$ combineMultipleModels_2d \
    --input example/nih3t3_generated_2d_2.txt \
    --geneXColName 0.5_0_0_0.5_x \
    --geneYColName 0.5_0_0_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --outputs example/first_example_2d_cellgroup1_1gauss \
    example/first_example_2d_cellgroup1_2gauss \
    example/first_example_2d_cellgroup1_1M_3gauss \
    --figure example/first_example_2d_cellgroup1_1-3gauss.png \
    --getPVal \
    --title "first example cell group 1 1,2,3 gauss"
```

The lines printed indicates that it uses only 282 independent samples (1 from the 1 Gaussian model, 264 from the 2 Gaussians model and 15 from the 3 Gaussians model).



1.4.3 Compare means from baredSC results

baredSC outputs can be used to evaluate the fold-change of expression between 2 conditions.

Inputs

We use the same inputs as in *previous input descriptions*.

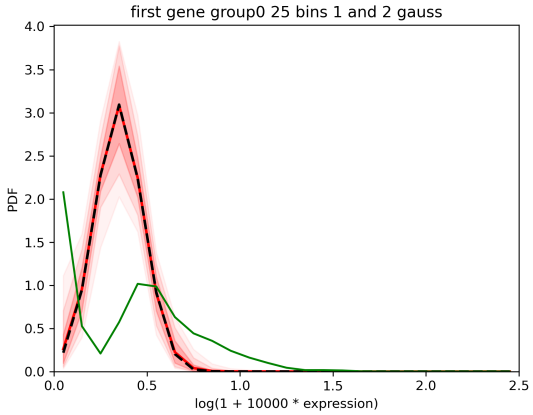
Run baredSC on each subpopulation

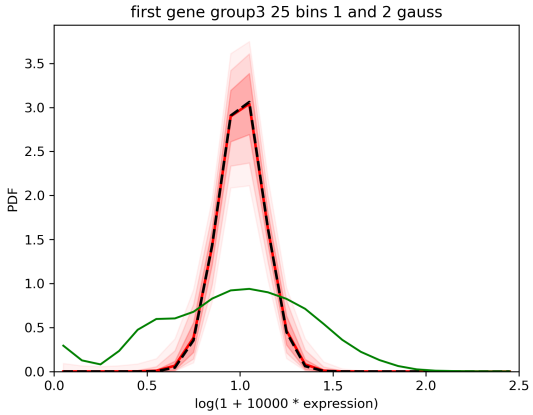
Let's focus on cells of group 0.0 and group 3.0 (they correspond to each of the 2 gaussians found previously). To increase the speed we will use less bins (`--nx 25`):

```
$ for group in 0 3; do
    for nnorm in 1 2; do
      baredSC_1d \
        --input example/nih3t3_generated_2d_2.txt \
        --metadata1ColName 0.5_0_0_0.5_group \
        --metadata1Values ${group}.0 \
        --geneColName 0.5_0_0_0.5_x \
        --output example/first_example_1d_group${group}_${nnorm}gauss_25_neff200 \
        --nnorm ${nnorm} --nx 25 --minNeff 200 \
        --figure example/first_example_1d_group${group}_${nnorm}gauss_25_neff200.png \
        --title "first gene ${nnorm} gauss group${group} 25 bins neff200" \
        --logevidence example/first_example_1d_group${group}_${nnorm}gauss_25_neff200_
↪logevid.txt
    done
    combineMultipleModels_1d \
      --input example/nih3t3_generated_2d_2.txt \
      --metadata1ColName 0.5_0_0_0.5_group \
      --metadata1Values ${group}.0 \
      --geneColName 0.5_0_0_0.5_x --nx 25 \
      --outputs example/first_example_1d_group${group}_1gauss_25_neff200 \
      example/first_example_1d_group${group}_2gauss_25_neff200 \
      --figure example/first_example_1d_group${group}_1-2gauss_25.png \
      --title "first gene group$group 25 bins 1 and 2 gauss"
  done
```

In one case, 100 000 samples were not sufficient.

We check the QC. We can now compare the results:



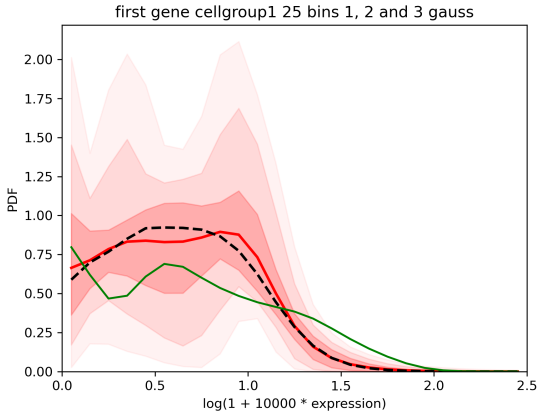


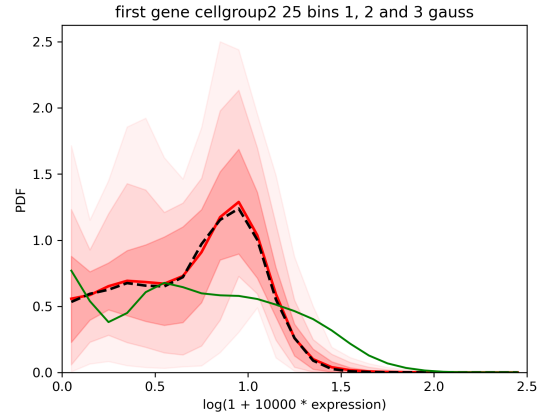
We can see that the tool fits relatively nicely the gaussians which were in inputs.

If we use another metadata which is just 300 and 500 random cells:

```
$ for group in 1 2; do
    for nnorm in 1 2 3; do
      baredSC_1d \
        --input example/nih3t3_generated_2d_2.txt \
        --metadata1ColName group \
        --metadata1Values group${group} \
        --geneColName 0.5_0_0_0.5_x \
        --output example/first_example_1d_cellgroup${group}_${nnorm}gauss_25_neff200 \
        --nnorm ${nnorm} --nx 25 --minNeff 200 \
        --figure example/first_example_1d_cellgroup${group}_${nnorm}gauss_25_neff200.png␣
↪\
        --title "first gene ${nnorm} gauss cellgroup${group}" \
        --logevidence example/first_example_1d_cellgroup${group}_${nnorm}gauss_25_
↪neff200_logevid.txt
    done
    combineMultipleModels_1d \
      --input example/nih3t3_generated_2d_2.txt \
      --metadata1ColName group \
      --metadata1Values group${group} \
      --geneColName 0.5_0_0_0.5_x --nx 25 \
      --outputs example/first_example_1d_cellgroup${group}_1gauss_25_neff200 \
      example/first_example_1d_cellgroup${group}_2gauss_25_neff200 \
      example/first_example_1d_cellgroup${group}_3gauss_25_neff200 \
      --figure example/first_example_1d_cellgroup${group}_1-3gauss_25.png \
      --title "first gene cellgroup$group 25 bins 1, 2 and 3 gauss"
  done
```

We see that the results are different in both groups but in both cases the confidence interval is quite large:





One of the output is *_*means.txt.gz*. Each line correspond to the value of the mean expression evaluated at each sample of the MCMC. It can be used to estimate:

- A confidence interval on the mean value (in the used axis log(1 + 10^4 * expression))
- A confidence interval on the fold change (delogged).

```
In [1]: import numpy as np
   ...: def delog(x):
   ...:     return(1e-4 * (np.exp(x) - 1))
   ...: my_quantiles = [0.5 - 0.6827 / 2, 0.5, 0.5 + 0.6827 / 2]
   ...: # Get data
   ...: means1 = np.genfromtxt('../example/first_example_1d_group0_1-2gauss_25_means.txt.
↪gz')
   ...: print(f'Values of mean in group0: {means1}')
   ...: means2 = np.genfromtxt('../example/first_example_1d_group3_1-2gauss_25_means.txt.
↪gz')
   ...: print(f'Values of mean in group3: {means2}')
   ...: # Shuffle means2
   ...: np.random.shuffle(means2)
   ...: print(f'Mean log expression in group0: {np.quantile(means1, my_quantiles)}')
   ...: print(f'Mean log expression in group3: {np.quantile(means2, my_quantiles)}')
   ...: fc = [delog(x1) / delog(x2) for x1, x2 in zip(means1, means2)]
   ...: print(f'Estimation of fold-change: {np.quantile(fc,  my_quantiles)}')
   ...:
```

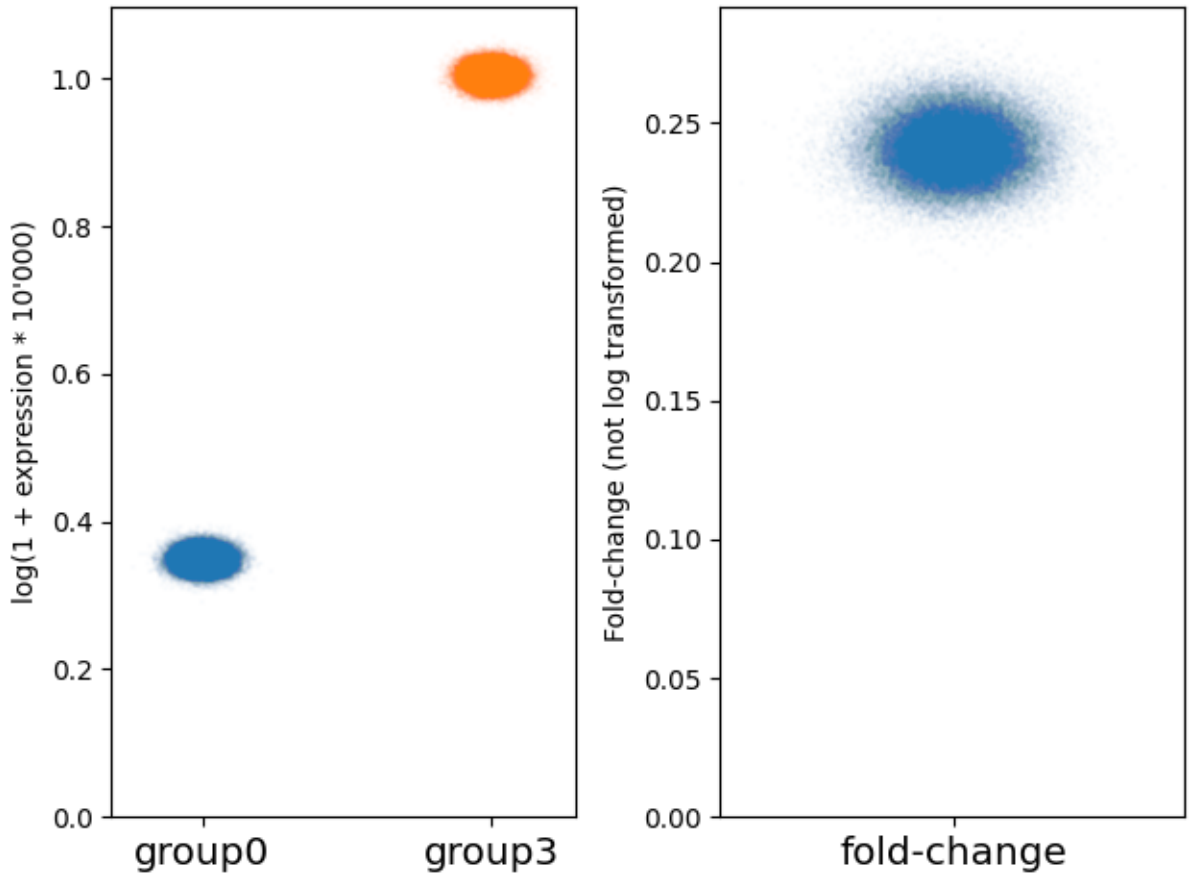
(continues on next page)

(continued from previous page)

```
Values of mean in group0: [0.35776993 0.35776993 0.35776993 ... 0.34672964 0.34672964 0.
↪38877004]
Values of mean in group3: [1.00746002 1.02085388 1.02085388 ... 1.00369734 1.01363153 1.
↪02256951]
Mean log expression in group0: [0.33838793 0.34953488 0.36060358]
Mean log expression in group3: [0.99479228 1.00628287 1.01751424]
Estimation of fold-change: [0.2310856  0.24111249 0.25135073]
```

We can use matplotlib to display the results graphically:

```
In [2]: import matplotlib.pyplot as plt
   ...: x1, x2 = np.random.normal(1, 0.1, len(means1)), np.random.normal(3, 0.1,␣
↪len(means2))
   ...: fig, axs = plt.subplots(1, 2)
   ...: axs[0].scatter(x1, means1, s=1, alpha=0.01)
   ...: axs[0].scatter(x2, means2, s=1, alpha=0.01)
   ...: axs[0].set_ylim(0, )
   ...: axs[0].set_ylabel('log(1 + expression * 10\'000)')
   ...: axs[0].set_xticks([1, 3])
   ...: axs[0].set_xticklabels(['group0', 'group3'], fontsize='x-large')
   ...: axs[1].scatter(x1[:len(fc)], fc, s=1, alpha=0.01)
   ...: axs[1].set_xticks([1])
   ...: axs[1].set_xticklabels(['fold-change'], fontsize='x-large')
   ...: axs[1].set_ylim(0, )
   ...: axs[1].set_ylabel('Fold-change (not log transformed)')
   ...: fig.tight_layout()
   ...:
```



On the first example with group0 and group3 where each is a different gaussian, we see a mean log expression around the expected 0.375 value in group 0, 1 in group 3. We see that the fold-change is around 25% (this is the real fold-change, not log transformed).

Now if we have a look to the subsamples of cells (300 and 500 cells) and perform the same analysis:

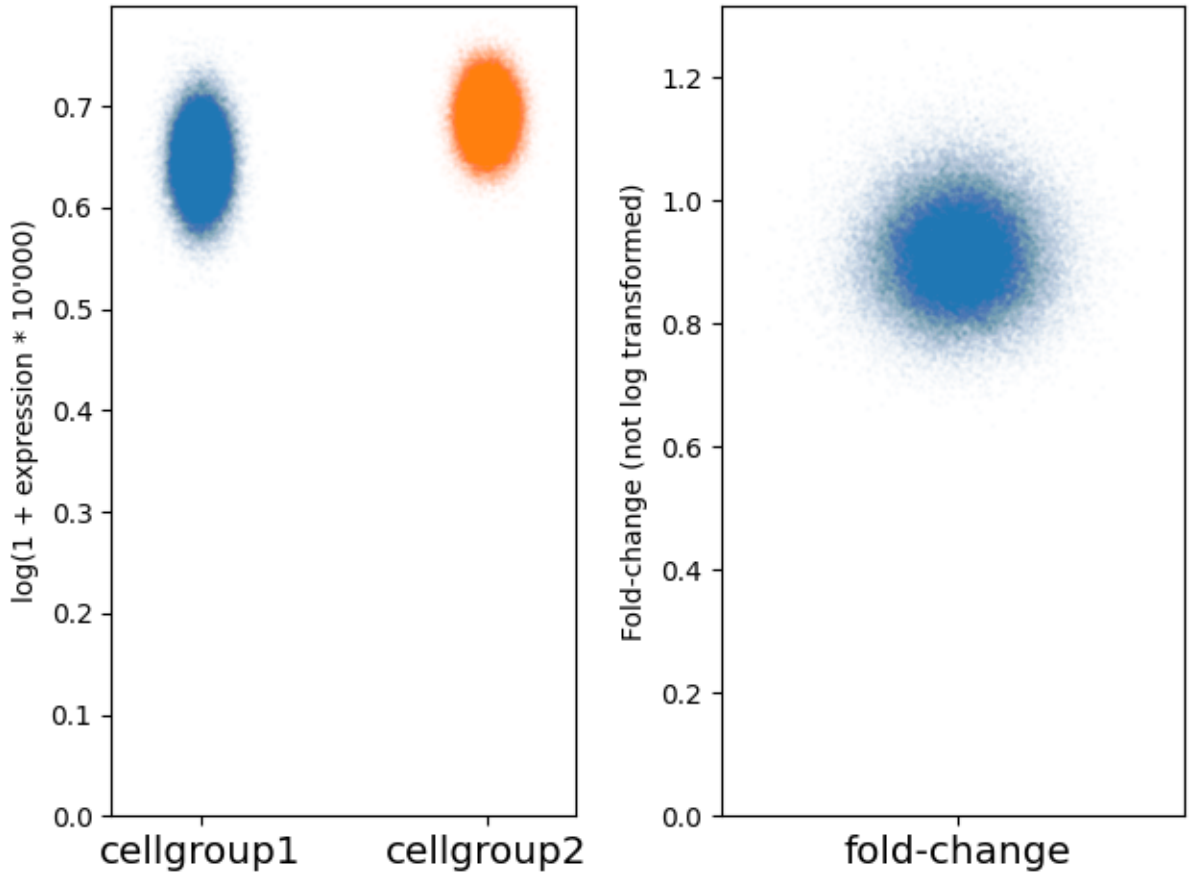
```
In [3]: # Get data
   ...: means1 = np.genfromtxt('../example/first_example_1d_cellgroup1_1-3gauss_25_means.
↪txt.gz')
   ...: print(f'Values of mean in cellgroup1: {means1}')
   ...: means2 = np.genfromtxt('../example/first_example_1d_cellgroup2_1-3gauss_25_means.
↪txt.gz')
   ...: print(f'Values of mean in cellgroup2: {means2}')
   ...: # Shuffle means2
   ...: np.random.shuffle(means2)
   ...: print(f'Mean log expression in cellgroup1: {np.quantile(means1, my_quantiles)}')
   ...: print(f'Mean log expression in cellgroup2: {np.quantile(means2, my_quantiles)}')
   ...: fc = [delog(x1) / delog(x2) for x1, x2 in zip(means1, means2)]
   ...: print(f'Estimation of fold-change: {np.quantile(fc,  my_quantiles)}')
   ...:
Values of mean in cellgroup1: [0.66035127 0.66035127 0.62547552 ... 0.62783495 0.
↪62783495 0.62783495]
Values of mean in cellgroup2: [0.68203872 0.69214061 0.71198453 ... 0.70698152 0.
↪70698152 0.70698152]
```

(continues on next page)

(continued from previous page)

```
Mean log expression in cellgroup1: [0.6161945  0.64620379 0.67630369]
Mean log expression in cellgroup2: [0.66763529 0.69110072 0.71426981]
Estimation of fold-change: [0.84307779 0.91221568 0.98619712]
```

```
In [4]: x1, x2 = np.random.normal(1, 0.1, len(means1)), np.random.normal(3, 0.1,␣
↪len(means2))
   ...: fig, axs = plt.subplots(1, 2)
   ...: axs[0].scatter(x1, means1, s=1, alpha=0.01)
   ...: axs[0].scatter(x2, means2, s=1, alpha=0.01)
   ...: axs[0].set_ylim(0, )
   ...: axs[0].set_ylabel('log(1 + expression * 10\'000)')
   ...: axs[0].set_xticks([1, 3])
   ...: axs[0].set_xticklabels(['cellgroup1', 'cellgroup2'], fontsize='x-large')
   ...: axs[1].scatter(x1[:len(fc)], fc, s=1, alpha=0.01)
   ...: axs[1].set_xticks([1])
   ...: axs[1].set_xticklabels(['fold-change'], fontsize='x-large')
   ...: axs[1].set_ylim(0, )
   ...: axs[1].set_ylabel('Fold-change (not log transformed)')
   ...: fig.tight_layout()
   ...:
```



Here, we can see that the fold-change is slightly below 1.

1.4.4 Change minScale

Inputs

We took total UMI counts from a real dataset of NIH3T3. We generated a example where 2 genes have the same distribution (2 gaussians, one of mean 0.375, scale 0.125 and another one of mean 1 and scale 0.1). Half of cells goes in each gaussian. The gene is called "0.5_0_0_0.5_x".

minScale

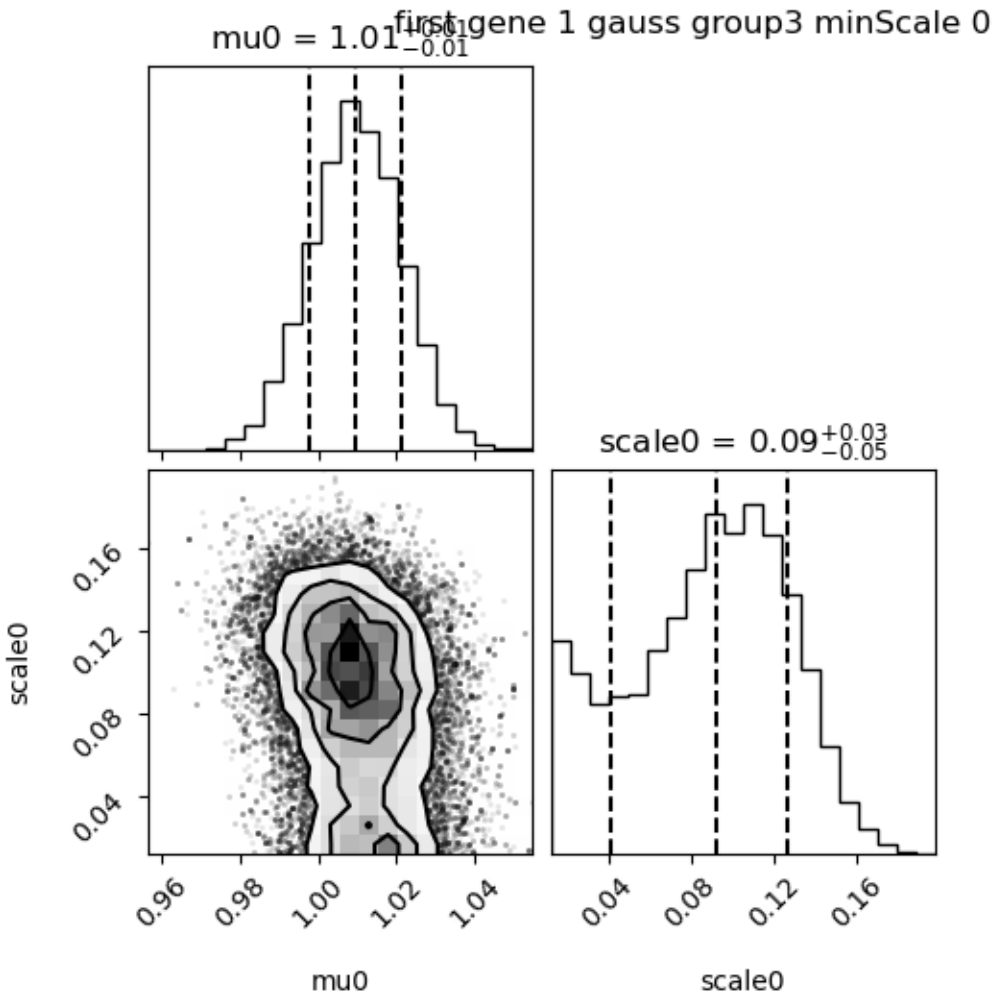
In our model, the prior on the scale of each Gaussian is that it must be above the `--minScale`. By default, this value is set to 0.1 because most of the time we don't have the resolution to go below. Here we will decrease this value to see how it affects the results

Run baredSC on the subpopulation with a scale of 0.1

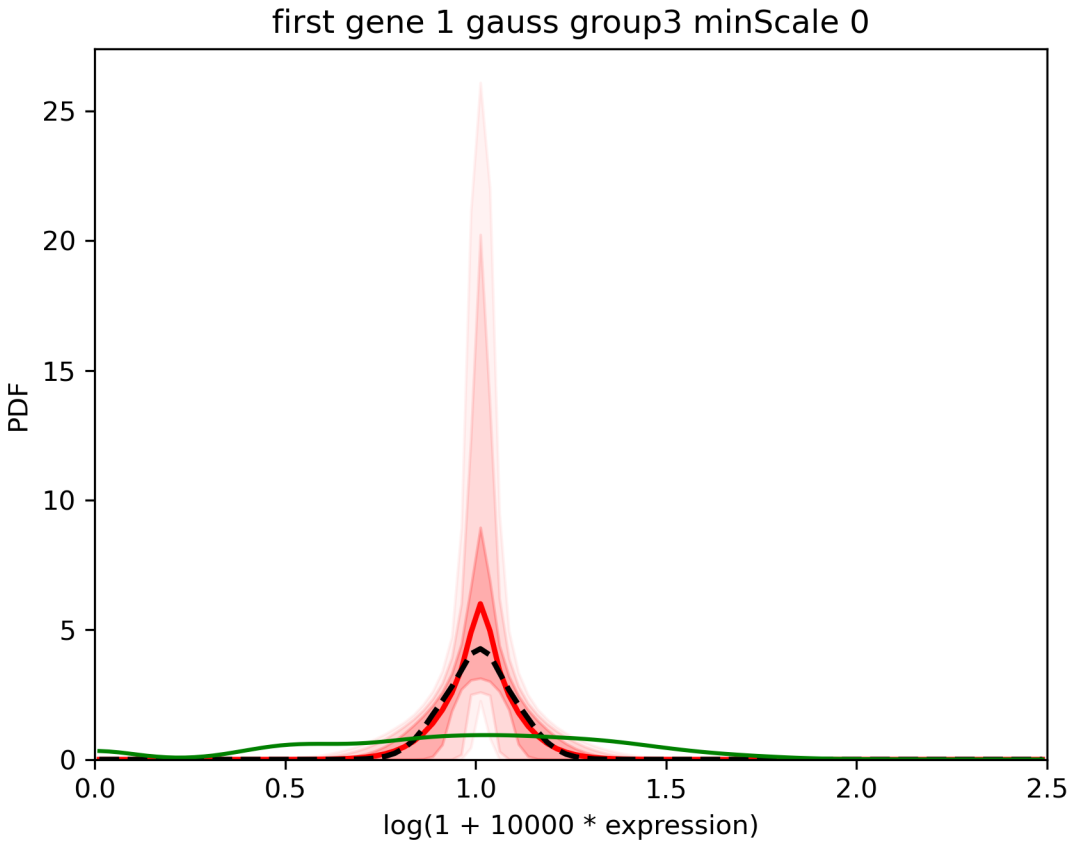
Let's focus on cells of group 3.0 (which corresponds to the second Gaussian of mean 1 and scale 0.1). We run baredSC but we put `--minScale` to 0. A minimum scale of 0 is not accepted by baredSC because it causes some issues so setting it to 0 will put the minimum value accepted by baredSC.

```
$ nnorm=1
$ baredSC_1d \
        --input example/nih3t3_generated_2d_2.txt \
        --metadata1ColName 0.5_0_0_0.5_group \
        --metadata1Values 3.0 \
        --geneColName 0.5_0_0_0.5_x \
        --output example/first_example_1d_group3_${nnorm}gauss_ms0 \
        --nnorm ${nnorm} --minScale 0 \
        --minNeff 200 \
        --figure example/first_example_1d_group3_${nnorm}gauss_ms0.png \
        --title "first gene ${nnorm} gauss group3 minScale 0"
```

First let's have a look to the corner plot:



We see that the median value for scale0 is 0.09 so very close to what was simulated. However, we also see some very small values of scale. As a consequence, when we look at the results:

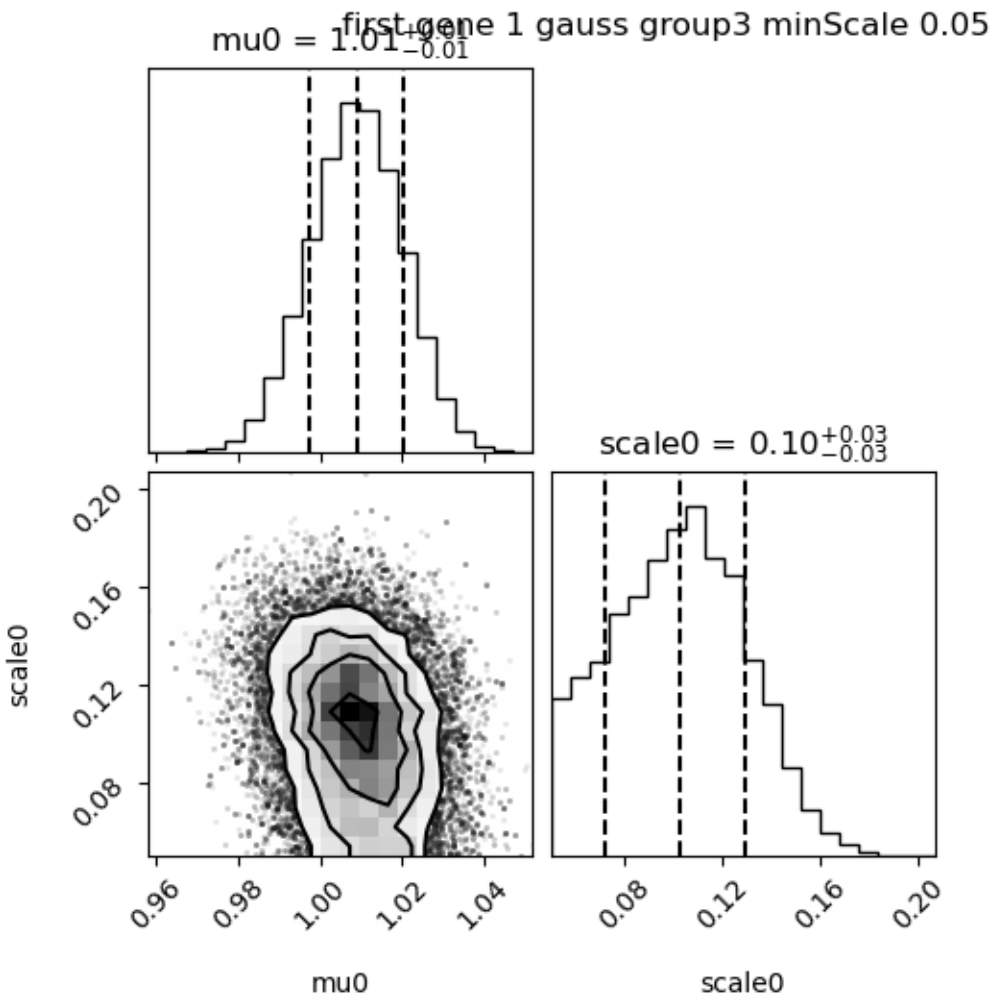


We see that the confidence interval is quite large and that the mean (red) is far from the median (dashed black).

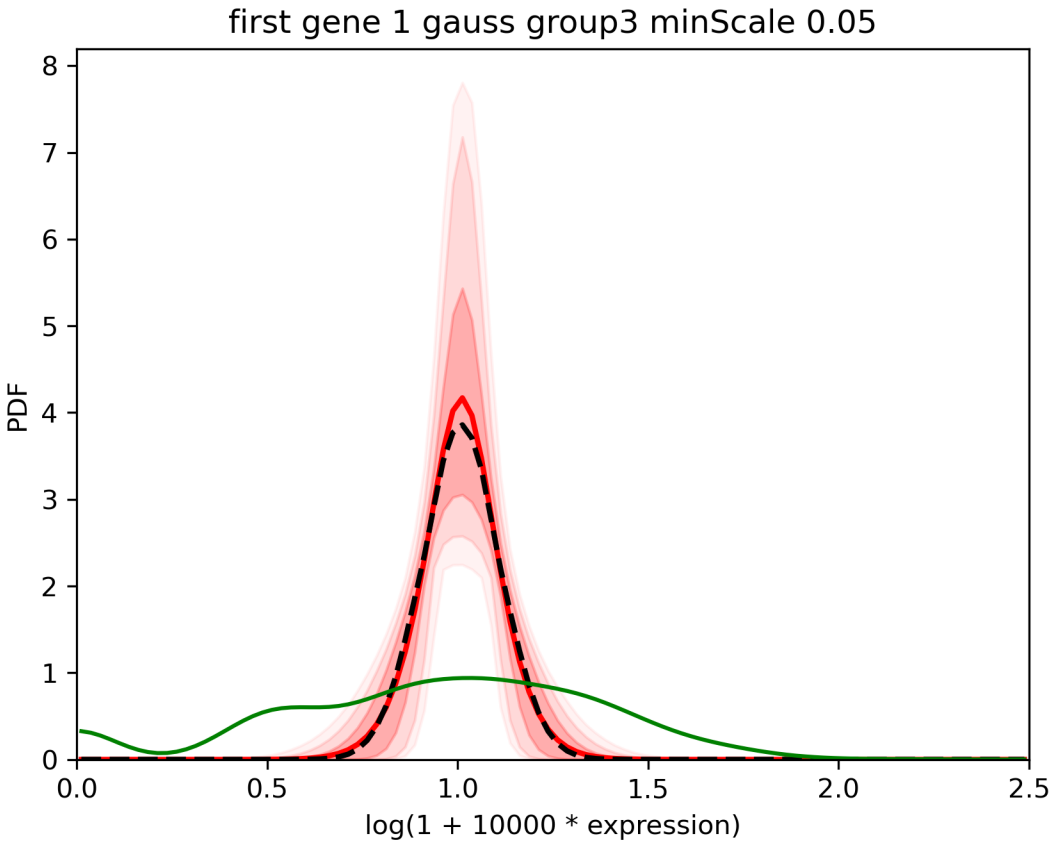
In such cases, it is reasonable to use a value for the minimum scale intermediate between the minimum value accepted by baredSC (0.0125) and the default value (0.1). For example, we can use 0.05:

```
$ nnorm=1
$ baredSC_1d \
    --input example/nih3t3_generated_2d_2.txt \
    --metadata1ColName 0.5_0_0_0.5_group \
    --metadata1Values 3.0 \
    --geneColName 0.5_0_0_0.5_x \
    --output example/first_example_1d_group3_${nnorm}gauss_ms0.05 \
    --nnorm ${nnorm} --minScale 0.05 \
    --minNeff 200 \
    --figure example/first_example_1d_group3_${nnorm}gauss_ms0.05.png \
    --title "first gene ${nnorm} gauss group3 minScale 0.05"
```

The corner plot shows that the median is still close to 0.1:



We see that the confidence interval is reduced large and that the mean (red) is closer from the median (dashed black).



1.4.5 Change the number of bins

- *Inputs*
- *Run baredSC in 2D*

baredSC can be relatively slow. A way to speed it is to decrease the number of bins.

Inputs

We took total UMI counts from a real dataset of NIH3T3. We generated a example where the PDF of the 2 genes is a 2D Gaussian. The mean on each axis and the scale on each axis is equal to 0.5 and the correlation value is also 0.5.

Run baredSC in 2D

By default baredSC_2d uses 50 bins in x and 50 bins in y. Let's run with default parameters.

```
$ nnorm=1
$ baredSC_2d \
    --input example/nih3t3_generated_second.txt \
    --geneXColName 1_0.5_0.5_0.5_0.5_0.5_x \
    --geneYColName 1_0.5_0.5_0.5_0.5_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --output example/second_example_2d_cellgroup1_${nnorm}gauss \
    --nnorm ${nnorm} \
    --figure example/second_example_2d_cellgroup1_${nnorm}gauss.png \
    --title "second example 2d cell group 1 ${nnorm} gauss"
```

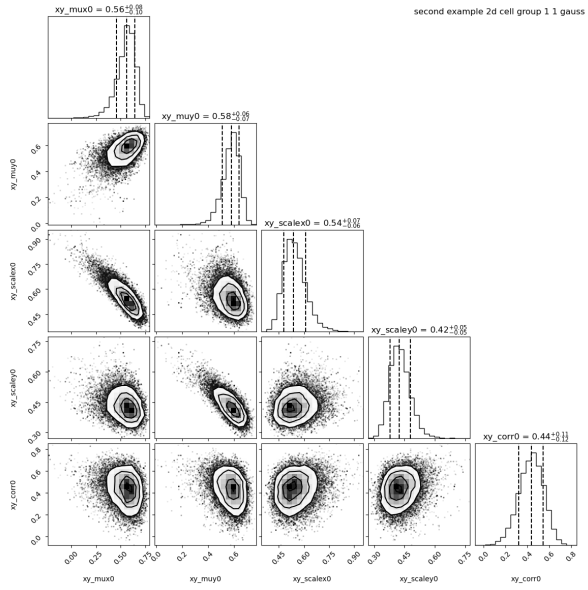
It took 11 minutes to run the MCMC and 3 minutes to compute the PDF.

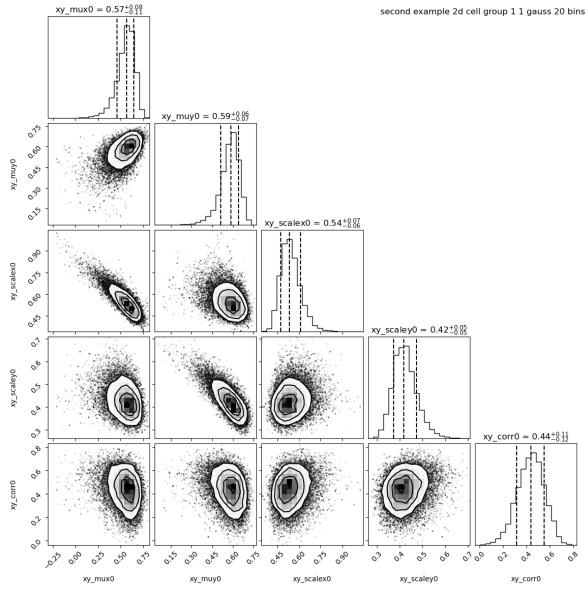
Let's try to reduce the number of bins using `--nx` and `--ny`:

```
$ nnorm=1
$ baredSC_2d \
    --input example/nih3t3_generated_second.txt \
    --geneXColName 1_0.5_0.5_0.5_0.5_0.5_x \
    --geneYColName 1_0.5_0.5_0.5_0.5_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --output example/second_example_2d_cellgroup1_${nnorm}gauss_nx20 \
    --nnorm ${nnorm} \
    --nx 20 --ny 20 \
    --figure example/second_example_2d_cellgroup1_${nnorm}gauss_nx20.png \
    --title "second example 2d cell group 1 ${nnorm} gauss 20 bins"
```

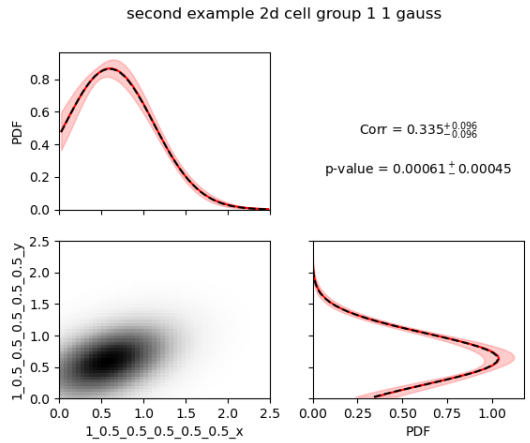
This time it took 3:32 minutes to compute MCMC and 30 seconds to get the PDF.

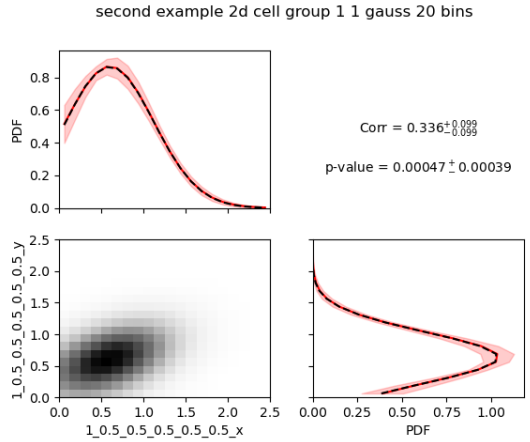
The parameters found are the same:





However, the image provided with 20 bins is much more pixelized:

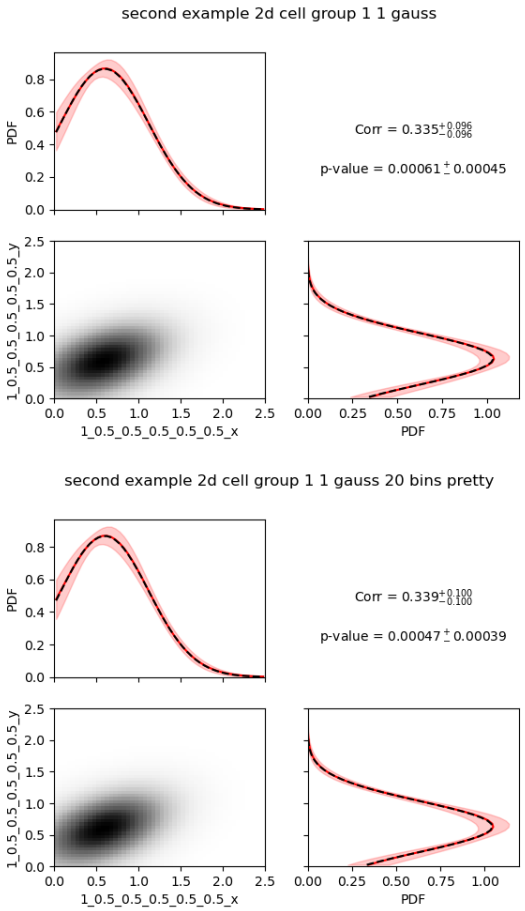




There is a way to render the plot prettier. However, you need to keep in mind that these pretty plots will not display the data as they have been used to compute the likelihood. In this example, the scale of the Gaussian is large enough that's why it gave the same results with both number of bins. We can set the number of bins to use in the plot with `--prettyBinsx` and `--prettyBinsy`.

```
$ nnorm=1
$ baredSC_2d \
    --input example/nih3t3_generated_second.txt \
    --geneXColName 1_0.5_0.5_0.5_0.5_0.5_x \
    --geneYColName 1_0.5_0.5_0.5_0.5_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --output example/second_example_2d_cellgroup1_${nnorm}gauss_nx20 \
    --nnorm ${nnorm} \
    --nx 20 --ny 20 --prettyBinsx 50 --prettyBinsy 50 \
    --figure example/second_example_2d_cellgroup1_${nnorm}gauss_nx20_pretty.png \
    --title "second example 2d cell group 1 ${nnorm} gauss 20 bins pretty"
```

It will use the `.npz` file generated with the last run to get the MCMC results.



Now they really look alike.

These options also exists in 1D.

1.4.6 Change scalePrior

- *Inputs*
- *Run baredSC in 2D*

baredSC_2d uses as a prior on the correlation value of each Gaussian a normal distribution. In order to reduce the number of false-positive (anti-)correlation detection. The scale of the normal distribution is set to 0.3. We show here the influence of this prior.

Inputs

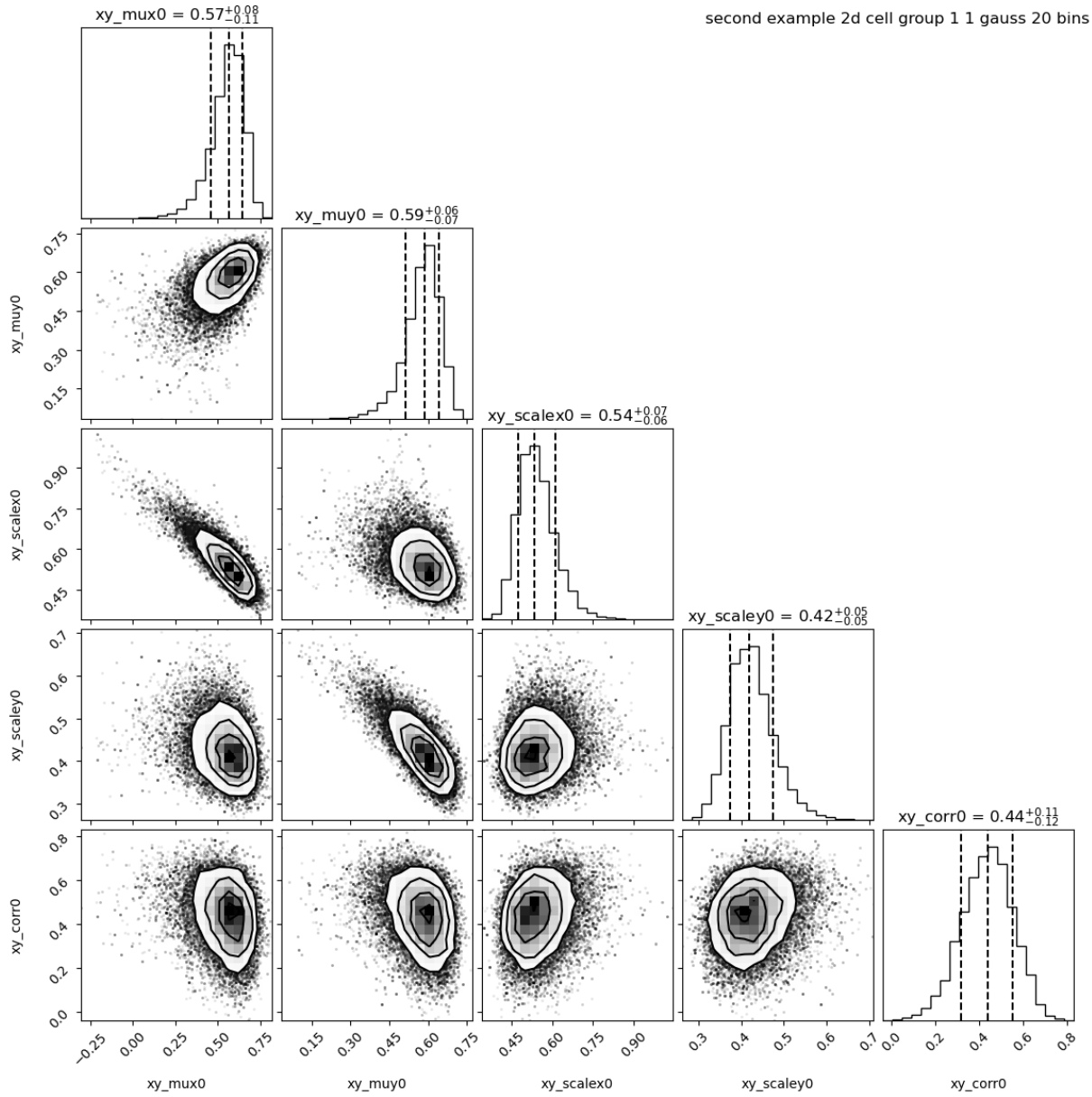
We took total UMI counts from a real dataset of NIH3T3. We generated a example where the PDF of the 2 genes is a 2D Gaussian. The mean on each axis and the scale on each axis is equal to 0.5 and the correlation value is also 0.5.

Run baredSC in 2D

By default baredSC_2d uses 50 bins in x and 50 bins in y. But to increase the speed we use only 20 bins:

```
$ nnorm=1
$ baredSC_2d \
    --input example/nih3t3_generated_second.txt \
    --geneXColName 1_0.5_0.5_0.5_0.5_0.5_x \
    --geneYColName 1_0.5_0.5_0.5_0.5_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --output example/second_example_2d_cellgroup1_${nnorm}gauss_nx20 \
    --nnorm ${nnorm} \
    --nx 20 --ny 20 \
    --figure example/second_example_2d_cellgroup1_${nnorm}gauss_nx20.png \
    --title "second example 2d cell group 1 ${nnorm} gauss 20 bins"
```

We see that the correlation found is 0.44 +/- 0.11.



Let see how this changes if we reduce the scale of the Normal distribution of the prior to 0.1

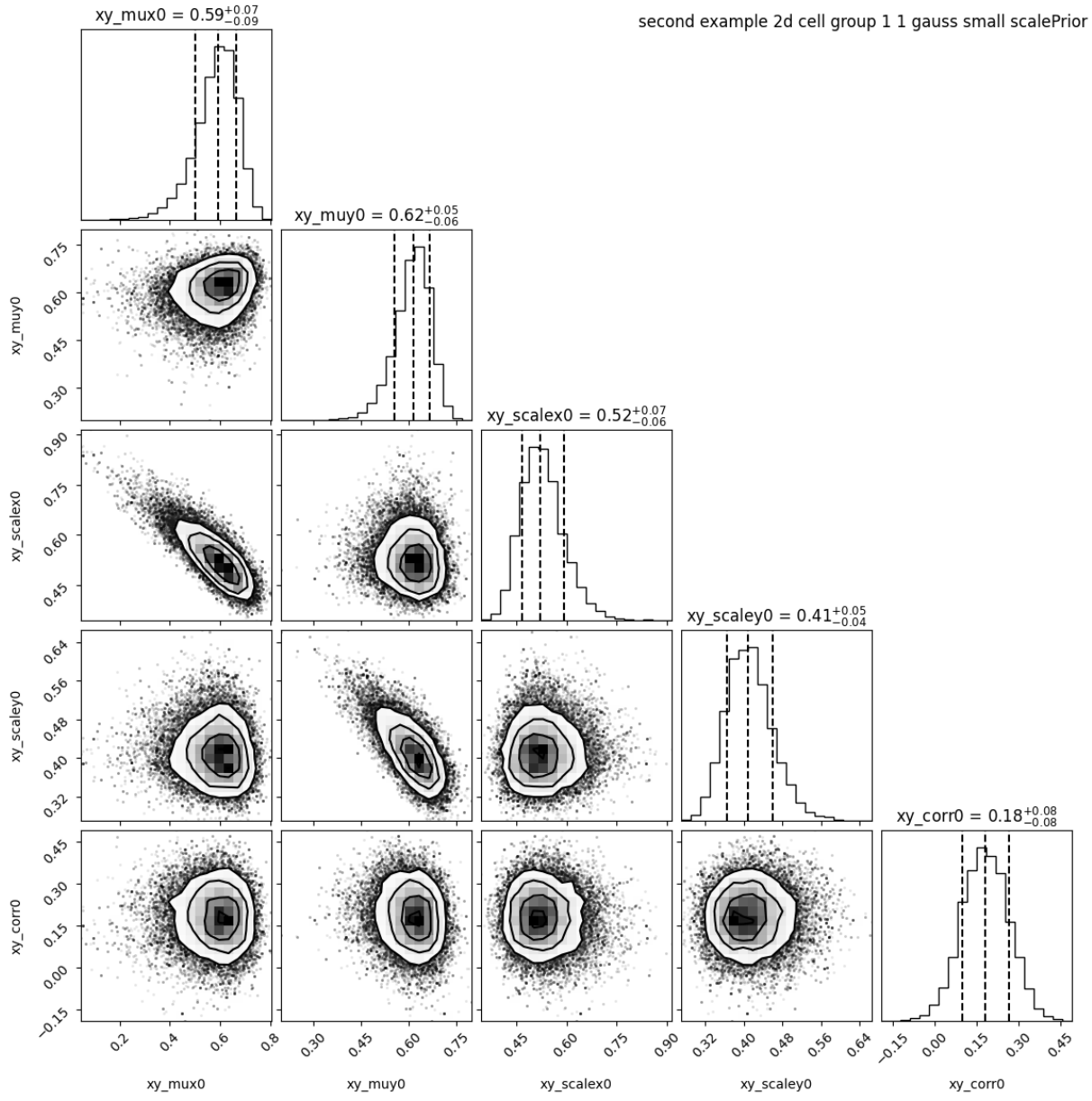
```
$ nnorm=1
$ baredSC_2d \
    --input example/nih3t3_generated_second.txt \
    --geneXColName 1_0.5_0.5_0.5_0.5_0.5_x \
    --geneYColName 1_0.5_0.5_0.5_0.5_0.5_y \
    --metadata1ColName group \
    --metadata1Values group1 \
    --output example/second_example_2d_cellgroup1_${nnorm}gauss_nx20_smallSP \
    --nnorm ${nnorm} \
    --nx 20 --ny 20 \
    --scalePrior 0.1 \
    --figure example/second_example_2d_cellgroup1_${nnorm}gauss_nx20_smallSP.png \
```

(continues on next page)

(continued from previous page)

```
    --title "second example 2d cell group 1 ${nnorm} gauss small scalePrior"
```

We see that the correlation drop to 0.18 +/- 0.08.



On the contrary, if we know that there is a correlation we can increase this value in order to remove the penalty on high correlation coefficient.

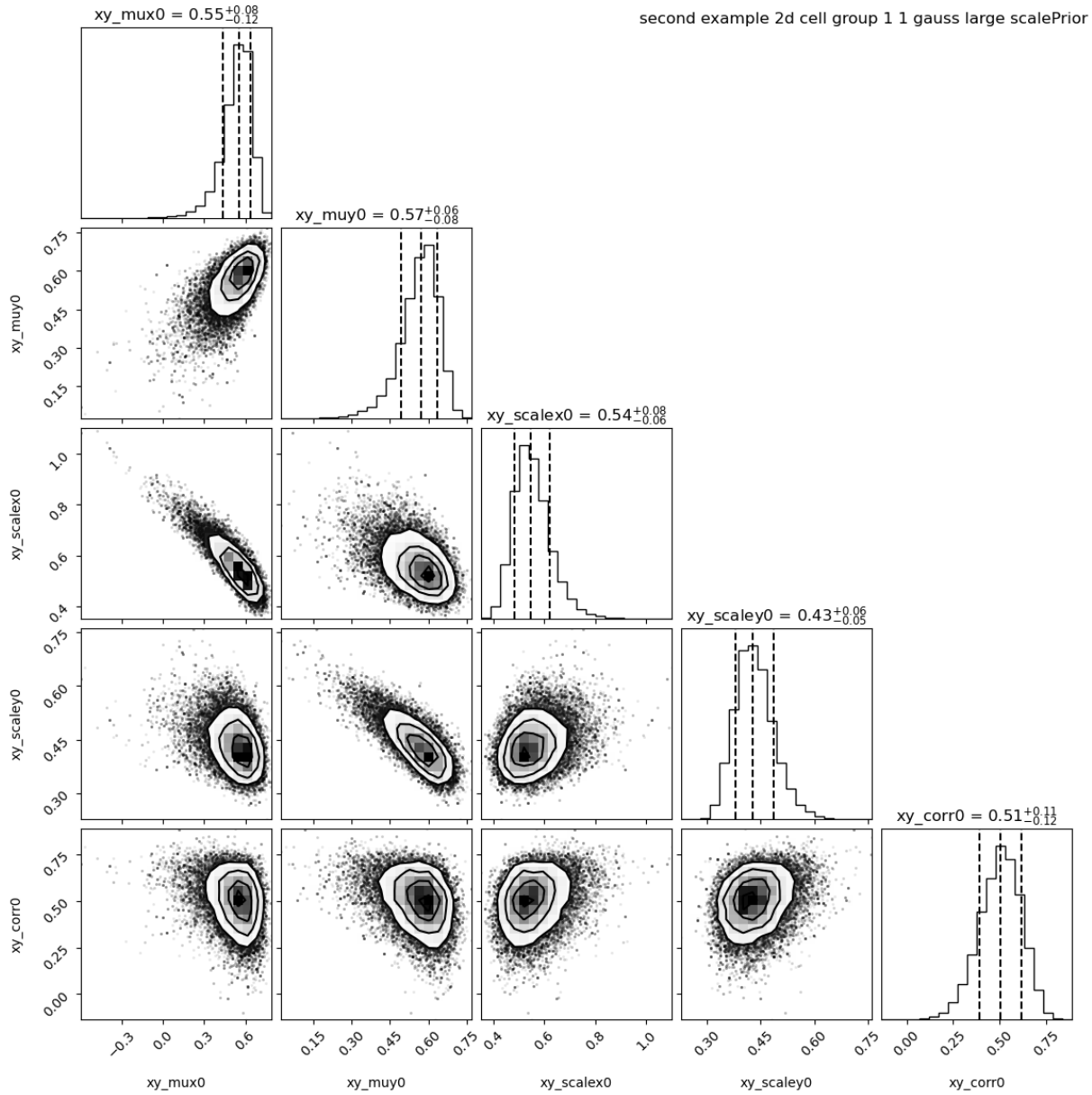
```
$ nnorm=1
$ baredSC_2d \
    --input example/nih3t3_generated_second.txt \
    --geneXColName 1_0.5_0.5_0.5_0.5_0.5_x \
    --geneYColName 1_0.5_0.5_0.5_0.5_0.5_y \
    --metadata1ColName group \
```

(continues on next page)

(continued from previous page)

```
    --metadata1Values group1 \
    --output example/second_example_2d_cellgroup1_${nnorm}gauss_nx20_largeSP \
    --nnorm ${nnorm} \
    --nx 20 --ny 20 \
    --scalePrior 3 \
    --figure example/second_example_2d_cellgroup1_${nnorm}gauss_nx20_largeSP.png \
    --title "second example 2d cell group 1 ${nnorm} gauss large scalePrior"
```

We see that the correlation is now at 0.51 +/- 0.11.



However, these settings may detect (anti-)correlations in situation where there is no, that's why we recommand the default value if you don't have any knowledge on the correlation you expect.

1.5 Releases

1.5.1 1.0.0

First release

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search