methylcheck Documentation

Release 0.8.4

FOXO Technologies, inc

Jul 08, 2022

Contents:

1 methylcheck is part of the methylsuite **3**

2 Methylsuite package components **5**

3 Installation **7**

4 Tutorials and Guides **9**

4.1 Loading processed methylation data and checking beta distributions . . . 9
4.1.1 Loading Beta Values and Metadata . . . 9
4.1.2 Loading Other Types of Data . . . 11
4.1.3 Loading Data from .csv Files . . . 15
4.1.4 Checking Beta Distributions . . . 17
4.2 Filtering Poor Quality Probes . . . 18
4.2.1 Available probe exclusion lists . . . 19
4.2.2 Filtering poor quality probes . . . 19
4.2.3 Filtering sex-linked probes and control probes . . . 22
4.3 Quality Control . . . 25
4.3.1 `controls_report` (a color-coded spreadsheet of control probe performance per sample) 25
4.3.2 Quality Control in the IDE . . . 28
4.3.3 Predicting Sex . . . 42
4.3.4 Report PDF tool . . . 43
4.4 Custom QC with pOOBAH Vales . . . 45
4.4.1 Load the Beta Values in a dataframe . . . 45
4.4.2 Load p-values in a dataframe . . . 48
4.4.3 Mask Beta values where probe fails . . . 49
4.4.4 Remove Samples based on Percent or Number of Failed Probes . . . 51
4.4.5 Drop out Probes with a Percentage of NaNs . . . 52
4.5 Outlier detection using Multidimensional Scaling (MDS) . . . 54
4.6 API Reference . . . 72
4.6.1 Loading Data . . . 73
4.6.2 `ReportPDF` Report Builder class . . . 73
4.6.3 Run QC pipeline . . . 73
4.6.4 filtering probes . . . 73
4.6.5 plotting functions . . . 73
4.6.6 sex prediction . . . 73
4.7 Release History . . . 73
4.7.1 v0.8.5 . . . 73

4.7.2 v0.8.4 . . . 73
4.7.3 v0.8.2 . . . 74
4.7.4 v0.8.1 . . . 74
4.7.5 v0.8.0 . . . 74
4.7.6 v0.7.9 . . . 74
4.7.7 v0.7.6 . . . 74
4.7.8 v0.7.5 . . . 75
4.7.9 v0.7.4 . . . 75
4.7.10 v0.7.3 . . . 75
4.7.11 v0.7.2 . . . 75
4.7.12 v0.7.1 . . . 76
4.7.13 v0.7.0 . . . 76
4.7.14 v0.6.4 . . . 76
4.7.15 v0.6.3 . . . 76
4.7.16 v0.6.2 . . . 76
4.7.17 v0.6.1 . . . 77
4.7.18 v0.6.0 . . . 77
4.7.19 v0.5.9 . . . 77
4.7.20 v0.5.7 . . . 77
4.7.21 v0.5.4 . . . 77
4.7.22 v0.5.2 . . . 77
4.7.23 v0.5.1 . . . 77
4.7.24 v0.5.0 . . . 77
4.7.25 v0.4.0 . . . 78

5 Indices and tables **79**

`methylcheck` is a Python-based package for filtering and visualizing Illumina methylation array data. The focus is on quality control. View on ReadTheDocs.

[3]:

IlmnID	6285625091_R05C01	7796806148_R03C02	7796806148_R01C02	628562
cg00035864	-0.514961	-2.015454	-1.958699	
cg00061679	-0.647032	0.616919	0.547593	
cg00063477	-0.639987	4.207996	3.884379	
cg00121626	-0.876430	-0.253725	-0.271588	
cg00223952	-4.638848	-4.387776	-4.630668	

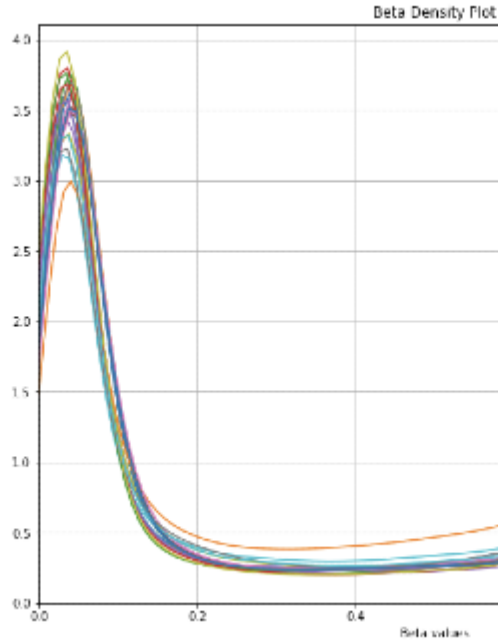
5 rows × 21 columns

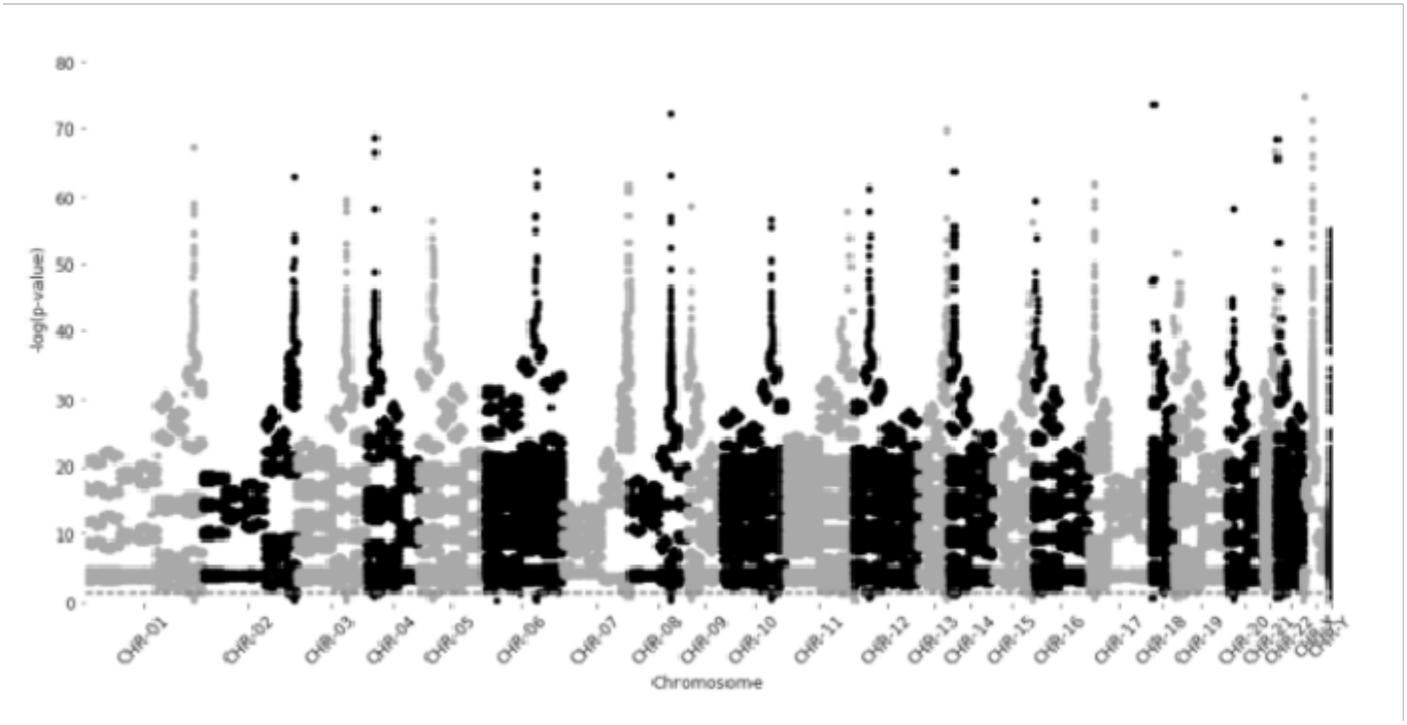
[4]:
```
meta_df = pd.read_pickle('sample_sheet_meta_data.pkl')
meta_df.head()
```

[4]:

	Cheez	BuffyCoat	Sentrix_ID	Sentrix_Position	Sample_Group	Sample_Name	Samp
0	1	0	6285625091	R05C01	None	8.24 CD34	
1	1	0	7796806148	R03C02	None	7.25 PMN	
2	1	0	7796806148	R01C02	None	7.25 PROS	
3	1	0	6285625091	R06C02	None	9.1 PMN	
4	2	0	6285625091	R03C01	None	8.10 CD19	

[19]:
```
methylcheck.sample_plot(beta_df)
```





CHAPTER 1

methylcheck is part of the methylsuite

`methylcheck` is part of the methylsuite of python packages that provide functions to process and analyze DNA methylation data from Illumina's Infinium arrays (27k, 450k, and EPIC, as well as mouse arrays). This package is focused on quality control for processed methylation data.

`methylcheck` functions are designed to work with a minimum of knowledge and specification required. But you can always override the "smart" defaults with custom settings if the default settings don't work for your data. The entire `methylsuite` is designed in this format: to offer ease of use while still maintaining flexibility for customization as needed.

CHAPTER 2

Methylsuite package components

You should install all three components, as they work together. The parts include:

- `methylprep`: for processing `idat` files or downloading GEO datasets from NIH. Processing steps include
  - infer type-I channel switch
  - NOOB (normal-exponential convolution on out-of-band probe data)
  - poobah (p-value with out-of-band array hybridization, for filtering low signal-to-noise probes)
  - qualityMask (to exclude historically less reliable probes)
  - nonlinear dye bias correction (AKA signal quantile normalization between red/green channels across a sample)
  - calculate beta-value, m-value, or copy-number matrix
  - large batch memory management, by splitting it up into smaller batches during processing
- `methylcheck`: (this package) for quality control (QC) and analysis, including
  - functions for filtering out unreliable probes, based on the published literature
    - Note that `methylprep process` will exclude a set of unreliable probes by default. You can disable that using the –no_quality_mask option from CLI.
  - sample outlier detection
  - array level QC plots of staining, bisulfite conversion, hybridization, extension, negative, non-polymorphic, target removal, and specificity
  - spreadsheet summary of control probe performance
  - data visualization functions based on `seaborn` and `matplotlib` graphic libraries.
  - predict sex of human samples from probes
  - interactive method for assigning samples to groups, based on array data, in a Jupyter notebook
- `methylize` provides more analysis and interpretation functions
  - differentially methylated probe statistics (between treatment and control samples)

- differentially methylated regions, with gene annotation from the UCSC Human Genome Browser
- volcano plots (to identify probes that are the most different)
- manhattan plots (to identify clusters of probes associated with genomic regions that are different)

CHAPTER 3

Installation

`methylcheck` maintains configuration files for your Python package manager of choice: pipenv or pip. Conda install is coming soon.

```
>>> pip install methylcheck
```

or if you want to install all three packages at once (recommended):

```
>>> pip install methylsuite
```

CHAPTER 4

Tutorials and Guides

If you are new to DNA methylation analysis, we recommend reading through this introduction from the `methylprep` documentation. Otherwise, you are ready to use `methylcheck` to:

- *load processed methylation data*
- *filter unreliable probes from your data*
- *run array-level quality control reports*
- *detect outlier samples*
- *predict the sex of human samples*

4.1 Loading processed methylation data and checking beta distributions

4.1.1 Loading Beta Values and Metadata

We will continue working with the dataset referenced in the `methylprep` general walkthrough: GSE147391. It was processed from the command line with the following command:

```
>>> python -m methylprep process -d <filepath> --all
```

Where `<filepath>` is the directory where the raw IDAT files are stored. We use `--all` to tell `methylprep` to give us all the possible output files, which include:

- beta_values.pkl
- poobah_values.pkl
- control_probes.pkl
- m_values.pkl
- noob_meth_values.pkl

- noob_unmeth_values.pkl
- meth_values.pkl
- unmeth_values.pkl
- sample_sheet_meta_data.pkl

As well as folders that contain the processed methylation data in .csv files.

[1]:
```
import methylcheck
from pathlib import Path
filepath = Path('/Users/patriciagirardi/tutorial/GPL21145')

betas = methylcheck.load(filepath)
betas.head()
```

```
Files: 100%|| 1/1 [00:00<00:00,  5.92it/s]
INFO:methylcheck.load_processed:loaded data (865859, 16) from 1 pickled files (0.222s)
```

[1]:

IlmnID	203163220027_R01C01	203163220027_R02C01	203163220027_R03C01
cg00000029	0.852	0.749	0.739
cg00000103	NaN	NaN	NaN
cg00000109	0.943	0.916	0.884
cg00000155	0.960	0.962	0.958
cg00000158	0.969	0.972	0.971

IlmnID	203163220027_R04C01	203163220027_R05C01	203163220027_R06C01
cg00000029	NaN	0.891	0.896
cg00000103	NaN	NaN	NaN
cg00000109	0.951	0.936	0.774
cg00000155	0.958	0.963	0.959
cg00000158	0.968	0.968	0.964

IlmnID	203163220027_R07C01	203163220027_R08C01	203175700025_R01C01
cg00000029	0.757	0.734	0.806
cg00000103	NaN	NaN	NaN
cg00000109	0.932	0.920	0.898
cg00000155	0.965	0.969	0.959
cg00000158	0.970	0.974	0.964

IlmnID	203175700025_R02C01	203175700025_R03C01	203175700025_R04C01
cg00000029	0.881	0.740	0.885
cg00000103	NaN	NaN	NaN
cg00000109	0.938	0.931	0.953
cg00000155	0.956	0.964	0.961
cg00000158	0.969	0.960	0.968

IlmnID	203175700025_R05C01	203175700025_R06C01	203175700025_R07C01
cg00000029	0.693	0.779	0.617
cg00000103	NaN	NaN	NaN
cg00000109	0.899	0.889	0.734
cg00000155	0.959	0.967	0.961
cg00000158	0.965	0.965	0.975

(continues on next page)

(continued from previous page)

```
            203175700025_R08C01
IlmnID
cg00000029                0.891
cg00000103                  NaN
cg00000109                0.892
cg00000155                0.960
cg00000158                0.966
```

You may also use `methylcheck.load_both()` to load both the beta values and metadata at the same time. Note that `methylcheck` expects the formatting used by `methylprep` in this command.

[2]:
```
df, meta = methylcheck.load_both(filepath)
meta.head()
```

```
INFO:methylcheck.load_processed:Found several meta_data files; attempting to match␣
↪each with its respective beta_values files in same folders.
WARNING:methylcheck.load_processed:Columns in sample sheet meta data files does not␣
↪match for these files and cannot be combined:['/Users/patriciagirardi/tutorial/
↪GPL21145/GSE147391_GPL21145_meta_data.pkl', '/Users/patriciagirardi/tutorial/
↪GPL21145/sample_sheet_meta_data.pkl']
INFO:methylcheck.load_processed:Multiple meta_data found. Only loading the first file.
INFO:methylcheck.load_processed:Loading 16 samples.
Files: 100%|| 1/1 [00:00<00:00,  5.98it/s]
INFO:methylcheck.load_processed:loaded data (865859, 16) from 1 pickled files (0.2s)
INFO:methylcheck.load_processed:meta.Sample_IDs match data.index (OK)
```

[2]:
```
        GSM_ID     Sample_Name    Sentrix_ID Sentrix_Position  \
2   GSM4429898   Grade II rep3  203163220027           R01C01
3   GSM4429899  Grade III rep1  203163220027           R02C01
12  GSM4429908   Grade IV rep5  203163220027           R03C01
15  GSM4429911   Grade IV rep8  203163220027           R04C01
6   GSM4429902   Grade II rep6  203163220027           R05C01

              source        histological diagnosis description  \
2   Resected glioma      Diffuse astrocytoma (II)      Glioma
3   Resected glioma  Anaplastic astrocytoma (III)      Glioma
12  Resected glioma             Glioblastoma (IV)      Glioma
15  Resected glioma             Glioblastoma (IV)      Glioma
6   Resected glioma        Oligodendroglioma (II)      Glioma

               Sample_ID
2   203163220027_R01C01
3   203163220027_R02C01
12  203163220027_R03C01
15  203163220027_R04C01
6   203163220027_R05C01
```

4.1.2 Loading Other Types of Data

Formats supported by `methylcheck.load()` are:

```
['beta_value', 'm_value', 'meth', 'meth_df', 'noob_df', 'sesame', 'beta_csv']
```

where 'beta_value' is the default.

[3]:
```
# the unnormalized probe values
(meth,unmeth) = methylcheck.load(filepath, format='meth_df')
meth.head()
```

```
100%|| 16/16 [00:00<00:00, 488.67it/s]
100%|| 16/16 [00:00<00:00, 1480.91it/s]
INFO:methylcheck.load_processed:(865859, 16) (865859, 16)
```

[3]:

IlmnID	203163220027_R01C01	203163220027_R02C01	203163220027_R03C01	\
cg00000029	933.0	874.0	845.0	
cg00000103	NaN	NaN	NaN	
cg00000109	2597.0	2428.0	2309.0	
cg00000155	3491.0	4397.0	3956.0	
cg00000158	5556.0	5534.0	5384.0	

IlmnID	203163220027_R04C01	203163220027_R05C01	203163220027_R06C01	\
cg00000029	528.0	1579.0	1243.0	
cg00000103	NaN	NaN	NaN	
cg00000109	3082.0	2542.0	2382.0	
cg00000155	4059.0	4314.0	4059.0	
cg00000158	4833.0	5381.0	6078.0	

IlmnID	203163220027_R07C01	203163220027_R08C01	203175700025_R01C01	\
cg00000029	893.0	741.0	1016.0	
cg00000103	NaN	NaN	NaN	
cg00000109	2710.0	1930.0	2029.0	
cg00000155	4403.0	4302.0	3865.0	
cg00000158	5678.0	4738.0	4938.0	

IlmnID	203175700025_R02C01	203175700025_R03C01	203175700025_R04C01	\
cg00000029	1272.0	1035.0	1423.0	
cg00000103	NaN	NaN	NaN	
cg00000109	2551.0	2385.0	3010.0	
cg00000155	3734.0	4996.0	5297.0	
cg00000158	4690.0	5658.0	6767.0	

IlmnID	203175700025_R05C01	203175700025_R06C01	203175700025_R07C01	\
cg00000029	917.0	904.0	743.0	
cg00000103	NaN	NaN	NaN	
cg00000109	2219.0	1716.0	2132.0	
cg00000155	3787.0	5216.0	4389.0	
cg00000158	6030.0	5780.0	6300.0	

IlmnID	203175700025_R08C01
cg00000029	1386.0
cg00000103	NaN
cg00000109	2158.0
cg00000155	4404.0
cg00000158	5050.0

[4]:
```
# noob-normalized probe values
(noob_meth,noob_unmeth) = methylcheck.load(filepath, format='noob_df')
```

(continues on next page)

(continued from previous page)

```
noob_meth.head()
```

```
100%|| 16/16 [00:00<00:00, 550.34it/s]
100%|| 16/16 [00:00<00:00, 1499.81it/s]
INFO:methylcheck.load_processed:(865859, 16), (865859, 16)
```

[4]:

IlmnID	203163220027_R01C01	203163220027_R02C01	203163220027_R03C01	\
cg00000029	1174.0	1123.0	973.0	
cg00000103	NaN	NaN	NaN	
cg00000109	3372.0	3003.0	2728.0	
cg00000155	4647.0	5488.0	4754.0	
cg00000158	7810.0	7100.0	6721.0	

IlmnID	203163220027_R04C01	203163220027_R05C01	203163220027_R06C01	\
cg00000029	NaN	1745.0	1355.0	
cg00000103	NaN	NaN	NaN	
cg00000109	3645.0	2950.0	2766.0	
cg00000155	4858.0	5390.0	5019.0	
cg00000158	5904.0	6912.0	7875.0	

IlmnID	203163220027_R07C01	203163220027_R08C01	203175700025_R01C01	\
cg00000029	1074.0	812.0	1342.0	
cg00000103	NaN	NaN	NaN	
cg00000109	3160.0	2185.0	2684.0	
cg00000155	5198.0	5042.0	5300.0	
cg00000158	6893.0	5625.0	6925.0	

IlmnID	203175700025_R02C01	203175700025_R03C01	203175700025_R04C01	\
cg00000029	1497.0	1286.0	1696.0	
cg00000103	NaN	NaN	NaN	
cg00000109	3191.0	2902.0	3526.0	
cg00000155	4875.0	6244.0	6430.0	
cg00000158	6278.0	7188.0	8478.0	

IlmnID	203175700025_R05C01	203175700025_R06C01	203175700025_R07C01	\
cg00000029	1018.0	956.0	820.0	
cg00000103	NaN	NaN	NaN	
cg00000109	2488.0	1883.0	2384.0	
cg00000155	4254.0	6116.0	4892.0	
cg00000158	7100.0	6859.0	7247.0	

IlmnID	203175700025_R08C01
cg00000029	1524.0
cg00000103	NaN
cg00000109	2368.0
cg00000155	5048.0
cg00000158	5887.0

[5]:
```
# m_values
m_values = methylcheck.load(filepath, format='m_value')
m_values.head()
```

```
Files: 100%|| 1/1 [00:00<00:00,  5.41it/s]
INFO:methylcheck.load_processed:loaded data (865859, 16) from 1 pickled files (0.22s)
```

[5]:

IlmnID	203163220027_R01C01	203163220027_R02C01	203163220027_R03C01	\
cg00000029	2.525	1.579	1.500	
cg00000103	NaN	NaN	NaN	
cg00000109	4.047	3.444	2.930	
cg00000155	4.582	4.681	4.528	
cg00000158	4.960	5.135	5.042	

IlmnID	203163220027_R04C01	203163220027_R05C01	203163220027_R06C01	\
cg00000029	NaN	3.028	3.109	
cg00000103	NaN	NaN	NaN	
cg00000109	4.269	3.875	1.775	
cg00000155	4.518	4.717	4.538	
cg00000158	4.928	4.909	4.748	

IlmnID	203163220027_R07C01	203163220027_R08C01	203175700025_R01C01	\
cg00000029	1.643	1.466	2.055	
cg00000103	NaN	NaN	NaN	
cg00000109	3.780	3.516	3.133	
cg00000155	4.797	4.960	4.564	
cg00000158	4.996	5.238	4.746	

IlmnID	203175700025_R02C01	203175700025_R03C01	203175700025_R04C01	\
cg00000029	2.890	1.508	2.940	
cg00000103	NaN	NaN	NaN	
cg00000109	3.912	3.748	4.358	
cg00000155	4.444	4.738	4.606	
cg00000158	4.958	4.568	4.920	

IlmnID	203175700025_R05C01	203175700025_R06C01	203175700025_R07C01	\
cg00000029	1.175	1.819	0.685	
cg00000103	NaN	NaN	NaN	
cg00000109	3.162	3.008	1.468	
cg00000155	4.547	4.885	4.641	
cg00000158	4.788	4.795	5.284	

IlmnID	203175700025_R08C01
cg00000029	3.027
cg00000103	NaN
cg00000109	3.045
cg00000155	4.567
cg00000158	4.844

Note: If you point to a folder with multiple batches of samples, of different array types, you'll get an error. All samples in the path you choose need to have the same number of probes (same array type). Here is the message you'll get in this scenario.

```
ValueError: all the input array dimensions for the concatenation axis must match␣
↪exactly,
but along dimension 0, the array at index 0 has size 226618 and the array at index 1␣
↪has size 244827
```

(continues on next page)

(continued from previous page)

This shouldn't happen if you download and process data using `methylprep` (which will automatically detect array types and create separate folders for each array type). If you didn't process with `methylprep`, the workaround is manually moving the files from different array-types into separate folders and loading them separately.

The `meth` format for `methylcheck.load()` returns a list of `SampleDataContainer` objects.

`sdc[0]._SampleDataContainer__data_frame` will give you a dataframe of the first `SampleDataContainer` in your list called `sdc`.

4.1.3 Loading Data from .csv Files

`methylcheck` assumes you want to load data the fastest way, using a single high-performance python3 pickled dataframe. But there are times when you want to load from CSV output files instead. One use case is where you want to examine probes that were filtered out by poobah (p-value probe detection). The CSVs contain this information, whereas the pickled dataframe has it removed by default.

If you wish to load the beta values from CSVs, point the function to the parent directory where your CSVs are and it will automatically go through that directory recursively to concatinate all of the beta value columns from each CSV present. Please note that this will take longer than reading it from the beta pickle.

This function will by default mask the beta values of failed probes with `NaN` (p<=0.05). If you wish to view your raw beta values without any influence by poobah, add the argument `no_poobah=True`.

[6]:
```
df = methylcheck.load(filepath, format='beta_csv')
df
```

```
Files:   0%|          | 0/16 [00:00<?, ?it/s]INFO:numexpr.utils:NumExpr defaulting to␣
↪8 threads.
Files: 100%|| 16/16 [00:20<00:00,  1.29s/it]
INFO:methylcheck.load_processed:merging...
100%|| 16/16 [00:00<00:00, 524.09it/s]
```

[6]:

IlmnID	203163220027_R02C01	203163220027_R06C01	203163220027_R08C01	\
cg00000029	0.749	0.896	0.734	
cg00000103	0.922	0.950	0.654	
cg00000109	0.916	0.774	0.920	
cg00000155	0.962	0.959	0.969	
cg00000158	0.972	0.964	0.974	
...	...	...	...	
rs9363764	0.544	0.544	0.592	
rs939290	0.055	0.591	0.974	
rs951295	0.557	0.032	0.566	
rs966367	0.966	0.468	0.028	
rs9839873	0.968	0.618	0.975	

IlmnID	203163220027_R03C01	203163220027_R07C01	203163220027_R04C01	\
cg00000029	0.739	0.757	NaN	
cg00000103	0.931	0.712	0.924	
cg00000109	0.884	0.932	0.951	
cg00000155	0.958	0.965	0.958	
cg00000158	0.971	0.970	0.968	
...	...	...	...	
rs9363764	0.062	0.058	0.431	

(continues on next page)

(continued from previous page)

```
rs939290                 0.051                0.965                0.809
rs951295                 0.515                0.955                0.966
rs966367                 0.956                0.499                0.036
rs9839873                0.642                0.671                0.962

            203163220027_R01C01  203163220027_R05C01  203175700025_R02C01  \
IlmnID
cg00000029               0.852                0.891                0.881
cg00000103               0.943                0.931                0.943
cg00000109               0.943                0.936                0.938
cg00000155               0.960                0.963                0.956
cg00000158               0.969                0.968                0.969
...                        ...                  ...                  ...
rs9363764                0.962                0.036                0.046
rs939290                 0.966                0.070                0.964
rs951295                 0.548                0.975                0.024
rs966367                 0.029                0.478                0.488
rs9839873                0.969                0.969                0.969

            203175700025_R06C01  203175700025_R08C01  203175700025_R03C01  \
IlmnID
cg00000029               0.779                0.891                0.740
cg00000103               0.920                0.731                0.809
cg00000109               0.889                0.892                0.931
cg00000155               0.967                0.960                0.964
cg00000158               0.965                0.966                0.960
...                        ...                  ...                  ...
rs9363764                0.562                0.046                0.054
rs939290                 0.578                0.601                0.535
rs951295                 0.073                0.556                0.035
rs966367                 0.483                0.526                0.034
rs9839873                0.613                0.669                0.967

            203175700025_R07C01  203175700025_R04C01  203175700025_R01C01  \
IlmnID
cg00000029               0.617                0.885                0.806
cg00000103               0.611                0.365                0.946
cg00000109               0.734                0.953                0.898
cg00000155               0.961                0.961                0.959
cg00000158               0.975                0.968                0.964
...                        ...                  ...                  ...
rs9363764                0.541                0.551                0.971
rs939290                 0.625                0.603                0.962
rs951295                 0.974                0.523                0.029
rs966367                 0.455                0.034                0.027
rs9839873                0.974                0.972                0.970

            203175700025_R05C01
IlmnID
cg00000029               0.693
cg00000103               0.934
cg00000109               0.899
cg00000155               0.959
cg00000158               0.965
...                        ...
rs9363764                0.966
rs939290                 0.965
```

(continues on next page)

(continued from previous page)

```
rs951295                  0.949
rs966367                  0.051
rs9839873                 0.050

[865918 rows x 16 columns]
```

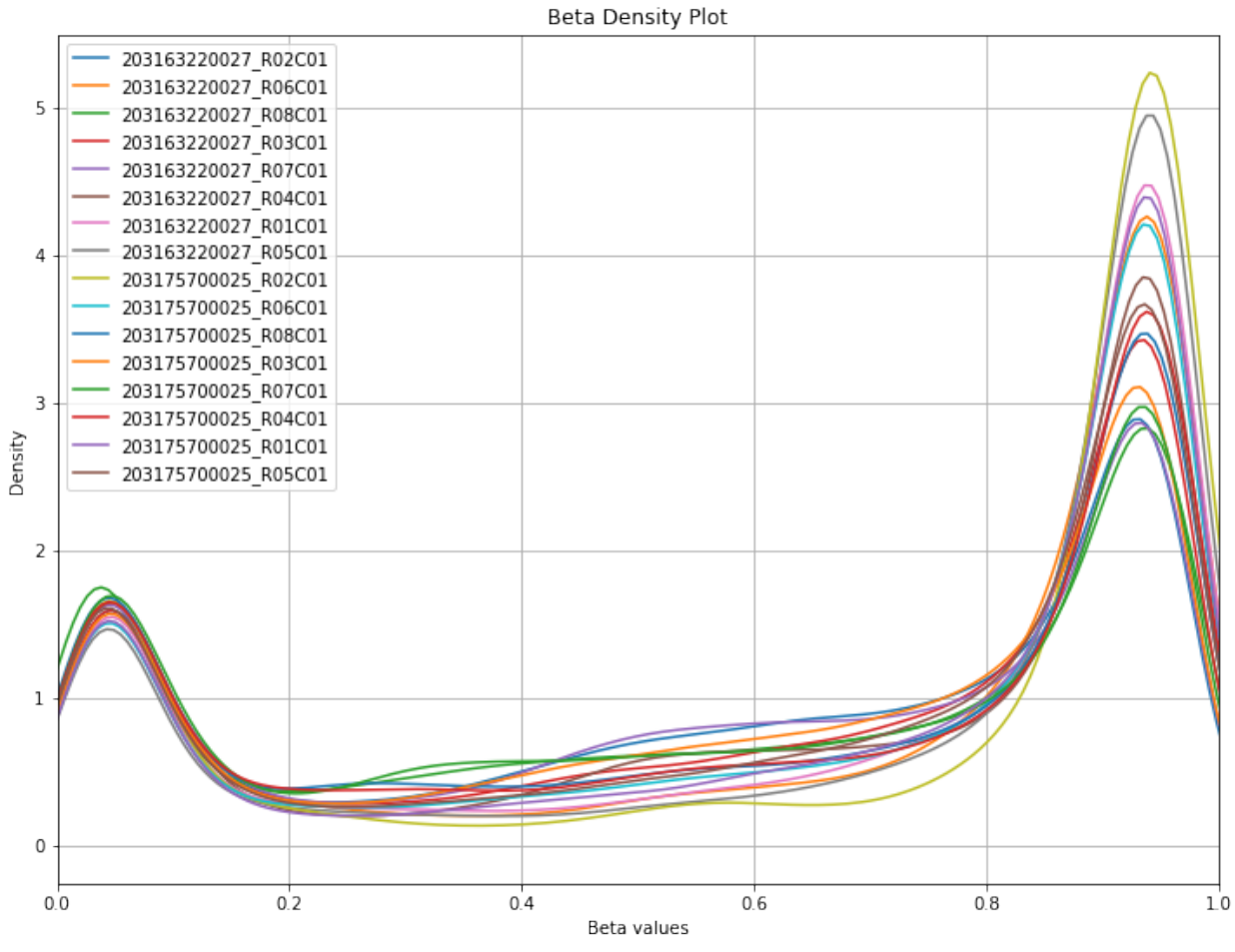
4.1.4 Checking Beta Distributions

One of the first steps users should take when loading their data is examining the beta distribution and ensuring that the expected bimodal distribution is present. Samples whose beta distributions don't fit the bimodal distribution are more likely to be poor quality samples that need to be filtered out (you may be expecting a different distribution based on what kind of data you're looking at, of course, but the standard is bimodal with peaks around 0 and 1).

`methylcheck` includes a beta density plot function that is similar to `seaborn`'s kde plot. If the sample size is small enough (<30), sample names will be included in the legend of the plot, so you may identify outlier samples easily.

[7]:
```
methylcheck.beta_density_plot(df)
```

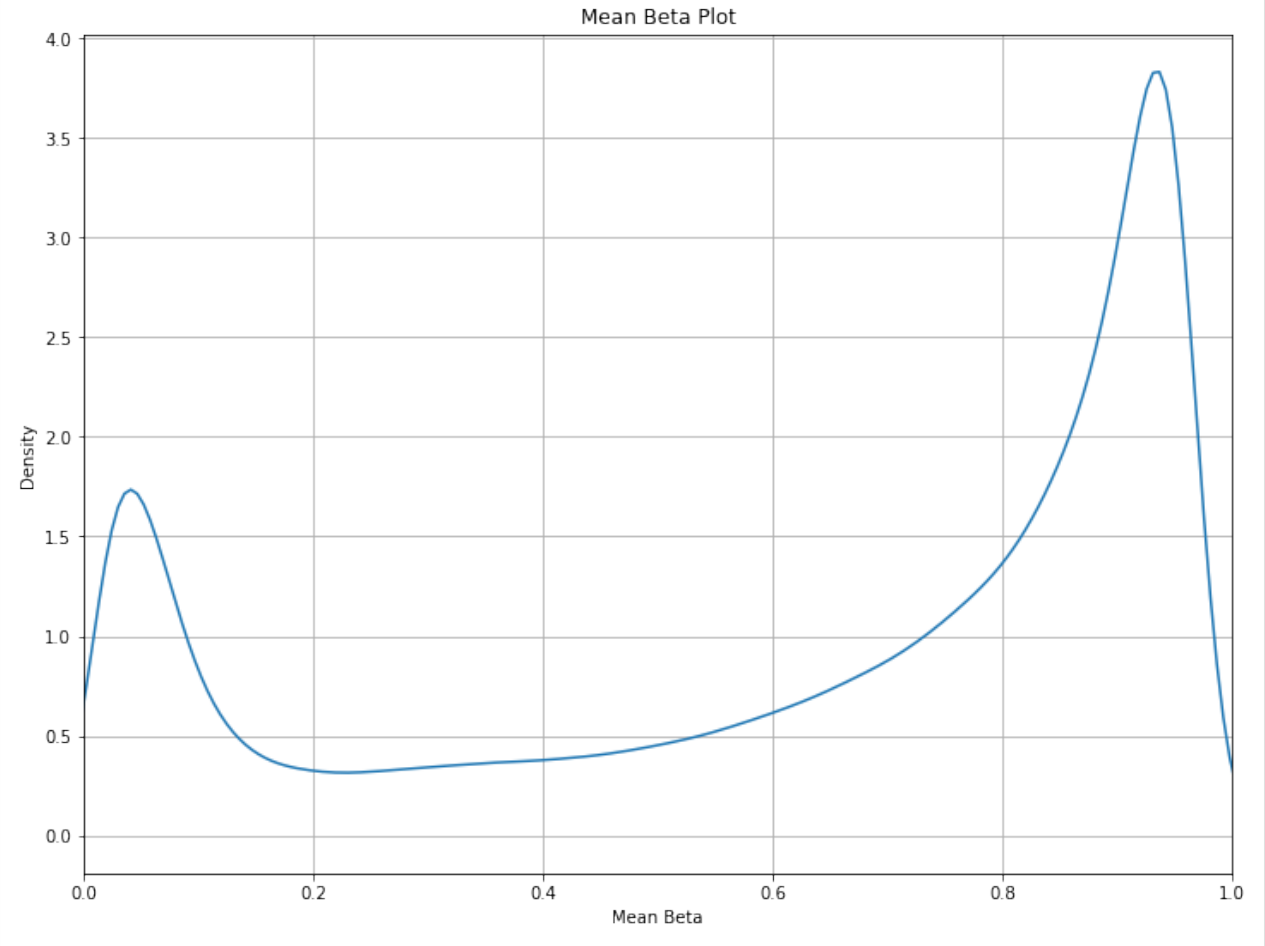
```
WARNING:methylcheck.samples.postprocessQC:Your data contains 7585 missing probe␣
↪values per sample, (121361 overall). For a list per sample, use verbose=True
```



This data looks relatively clean! Most data on GEO should be high quality, but it always pays to check.

You may also want to get an idea of the mean beta distribution (useful for comparing multiple datasets, or for looking at a dataset before/after quality control measures.) Check the filtering probes section for examples on comparing beta distributions and what filtering can do to your data's mean beta distribution.

[8]:
```
methylcheck.mean_beta_plot(df)
```



4.2 Filtering Poor Quality Probes

Note: Separate from any filtering that `methylcheck` does, the qualityMask step of the `methylprep` processing pipeline excludes a list of probes that are historically poor quality (`SeSAMe` masks the same list of probes, which are from the Zhou 2016 paper [linked below]). This can be turned off in `run_pipeline` by specifying `quality_mask=False` if desired. There are several critera for exclusion of probes that `methylcheck` offers. These are designed to be in line with past research that has identified "sketchy" probes. Areas that have polymorphisms, cross-hybridization, repeat sequence elements, or base color changes can affect probe quality. `methylcheck`'s `list_problem_probes()` function returns a list of probes excluded based on literature. For each array type, the publication list is as follows:

- 450k: 'Chen2013,' 'Price2013', 'Zhou2016', 'Naeem2014', 'DacaRoszak2015'
- EPIC: 'Zhou2016', 'McCartney2016'

The articles are linked above, for any users that would like more detail on what probes are considered problematic and what criteria the authors have used to identify them. The lists are also formatted in the acceptable input style for the `list_problem_probes` function. See below for examples of its usage.

[1]:
```
import methylcheck
from pathlib import Path
filepath = Path('/Users/patriciagirardi/tutorial/GPL21145')
```

4.2.1 Available probe exclusion lists

We've imported lists of methylation probes that various researchers have previously deemed to be "sketchy." You can use `methylcheck` to remove these probes – by referring to the list by the publication's first author name, or by the reason these probes should be excluded from analysis.

[2]:
```
# this code will print the criteria reason (either a publication or a type of issue, like Polymorphism)
# as well as the number of probes excluded for that reason

criteria = ['Chen2013', 'Price2013', 'Naeem2014', 'DacaRoszak2015','Polymorphism',
            'CrossHybridization', 'BaseColorChange', 'RepeatSequenceElements']
EPIC_criteria = ['McCartney2016', 'Zhou2016', 'Polymorphism', 'CrossHybridization', 'BaseColorChange', 'RepeatSequenceElements']

print('450k probe exclusion criteria and number of probes excluded:')
for crit in criteria:
    print(crit, '--', len(methylcheck.list_problem_probes('450k', [crit])))

print('\nEPIC probe exclusion criteria and number of probes excluded:')
for crit in EPIC_criteria:
    print(crit, '--', len(methylcheck.list_problem_probes('EPIC', [crit])))
```

```
450k probe exclusion criteria and number of probes excluded:
Chen2013 -- 265410
Price2013 -- 213246
Naeem2014 -- 128695
DacaRoszak2015 -- 89678
Polymorphism -- 289952
CrossHybridization -- 92524
BaseColorChange -- 359
RepeatSequenceElements -- 96631

EPIC probe exclusion criteria and number of probes excluded:
McCartney2016 -- 326267
Zhou2016 -- 178671
Polymorphism -- 346033
CrossHybridization -- 108172
BaseColorChange -- 406
RepeatSequenceElements -- 0
```

[3]:
```
# users may also get the list of probe names that are excluded for any of the criteria
methylcheck.list_problem_probes(array='epic', criteria=['Polymorphism'])[0:5]
```

[3]:
```
['cg14670079', 'cg26778521', 'cg04785903', 'cg15989966', 'cg21932343']
```

4.2.2 Filtering poor quality probes

[4]:
```
# leave criteria undefined to list all problem probes for that array type
sketchy_probes_list = methylcheck.list_problem_probes(array='epic')
```

[5]:
```
df = methylcheck.load(filepath)

# with the sketchy_probes_list defined above, we can use methylcheck.exclude_probes()␣
↪to remove all the unwanted probes
excluded_df = methylcheck.exclude_probes(df, sketchy_probes_list)
excluded_df.head()
```

```
Files: 100%|| 1/1 [00:00<00:00,  4.75it/s]
INFO:methylcheck.load_processed:loaded data (865859, 16) from 1 pickled files (0.253s)
```

```
Of {len(df.index)} probes, {post_overlap} matched, yielding {len(df.index)-post_
↪overlap} probes after filtering.
```

[5]:

IlmnID	203163220027_R01C01	203163220027_R02C01	203163220027_R03C01	\
cg00000029	0.852	0.749	0.739	
cg00000109	0.943	0.916	0.884	
cg00000165	0.237	0.733	0.374	
cg00000221	0.943	0.937	0.939	
cg00000236	0.922	0.934	0.919	

IlmnID	203163220027_R04C01	203163220027_R05C01	203163220027_R06C01	\
cg00000029	NaN	0.891	0.896	
cg00000109	0.951	0.936	0.774	
cg00000165	0.639	0.397	0.564	
cg00000221	0.926	0.942	0.945	
cg00000236	0.881	0.926	0.882	

IlmnID	203163220027_R07C01	203163220027_R08C01	203175700025_R01C01	\
cg00000029	0.757	0.734	0.806	
cg00000109	0.932	0.920	0.898	
cg00000165	0.486	0.194	0.163	
cg00000221	0.938	0.933	0.949	
cg00000236	0.927	0.774	0.937	

IlmnID	203175700025_R02C01	203175700025_R03C01	203175700025_R04C01	\
cg00000029	0.881	0.740	0.885	
cg00000109	0.938	0.931	0.953	
cg00000165	0.557	0.531	0.856	
cg00000221	0.936	0.934	0.949	
cg00000236	0.957	0.932	0.952	

IlmnID	203175700025_R05C01	203175700025_R06C01	203175700025_R07C01	\
cg00000029	0.693	0.779	0.617	
cg00000109	0.899	0.889	0.734	
cg00000165	0.171	0.173	0.865	
cg00000221	0.944	0.937	0.942	
cg00000236	0.923	0.938	0.932	

IlmnID	203175700025_R08C01
cg00000029	0.891
cg00000109	0.892
cg00000165	0.581
cg00000221	0.934

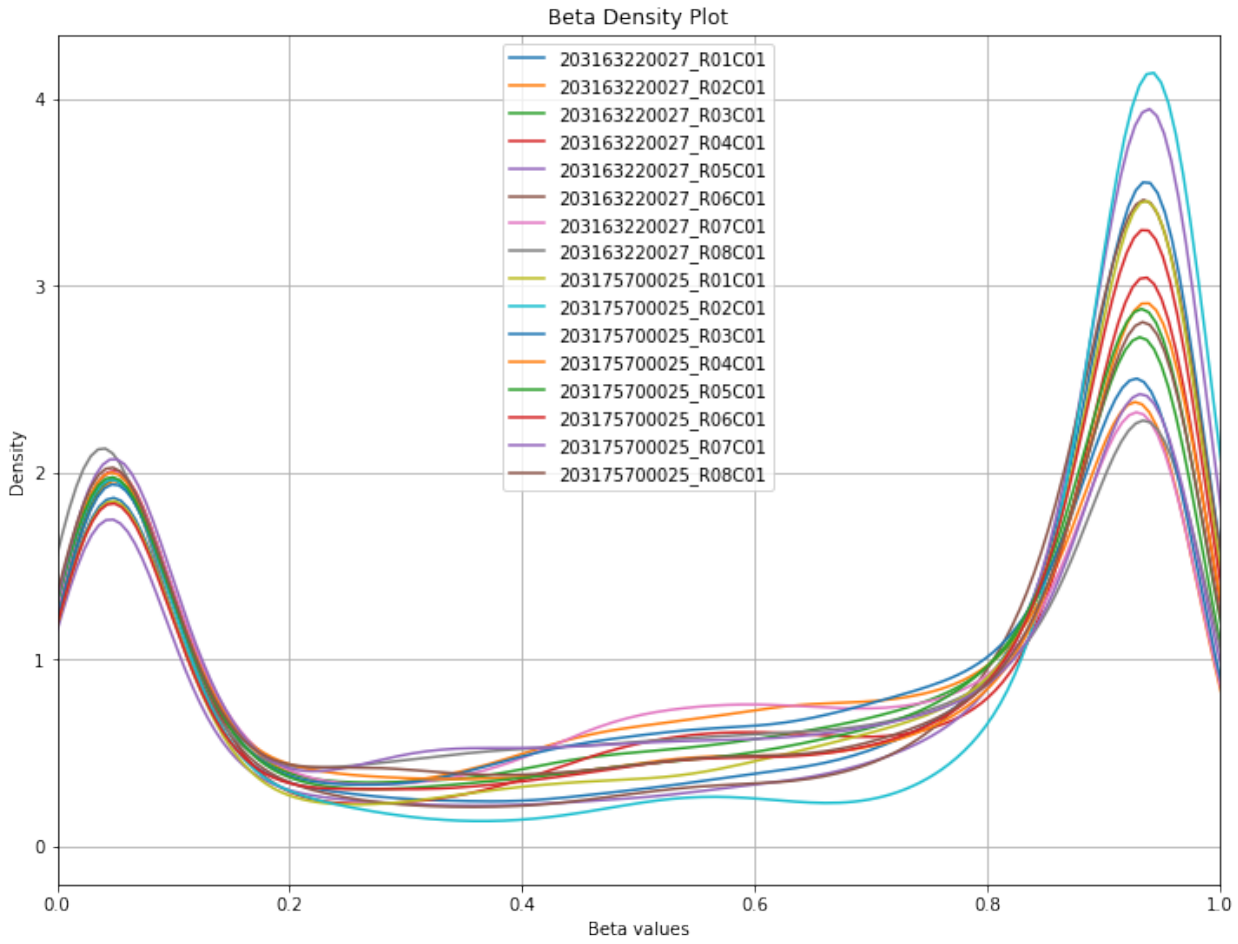
(continues on next page)

(continued from previous page)

```
cg00000236                0.951
```

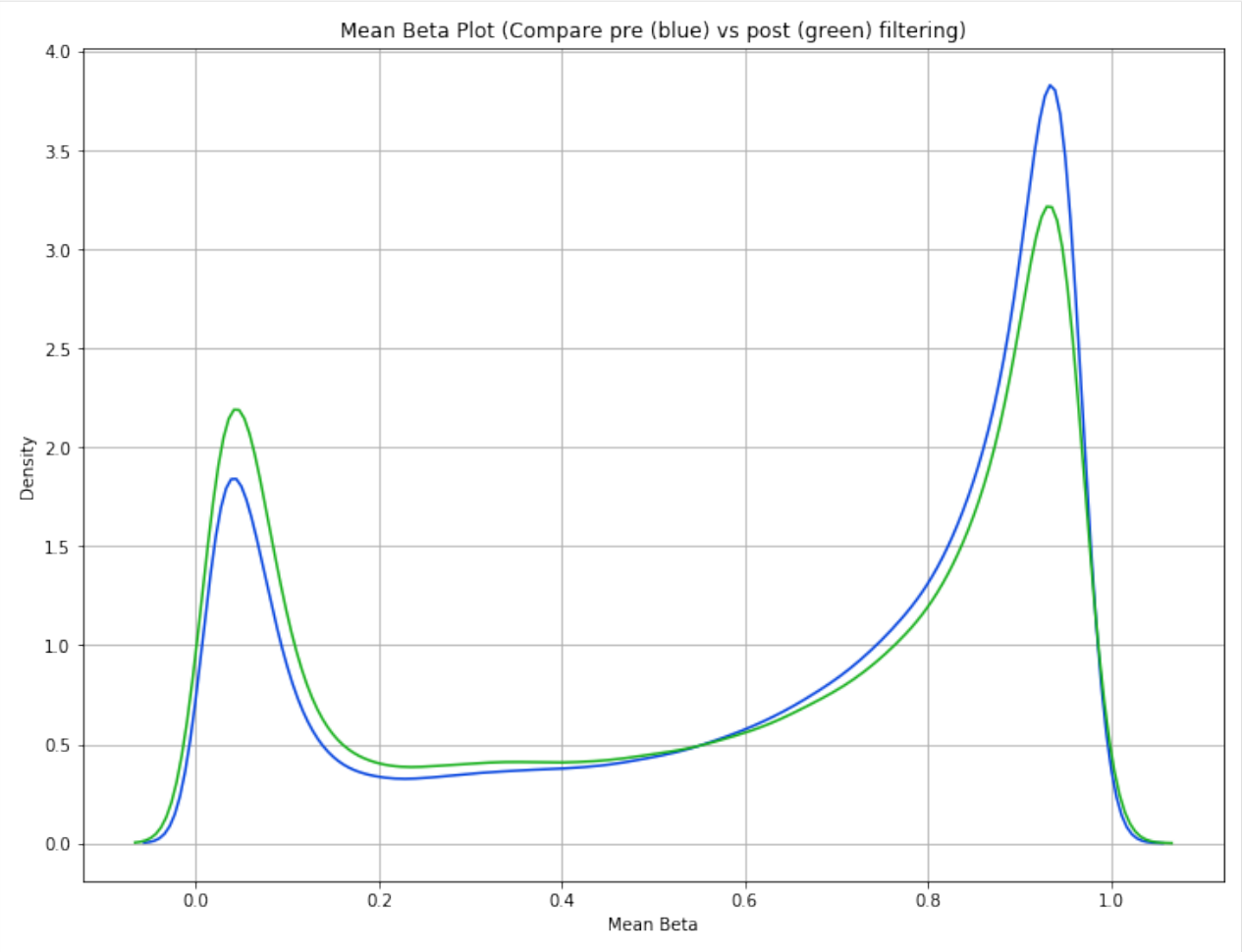
[6]:
```
methylcheck.beta_density_plot(excluded_df)
```

```
WARNING:methylcheck.samples.postprocessQC:Your data contains 6103 missing probe␣
↪values per sample, (97656 overall). For a list per sample, use verbose=True
INFO:numexpr.utils:NumExpr defaulting to 8 threads.
```



To get a sense for how this affects the data, users may want to compare the mean beta density distribution before and after probe filtering.

[7]:
```
methylcheck.mean_beta_compare(df, excluded_df)
```



4.2.3 Filtering sex-linked probes and control probes

Other probe types that are often filtered out are sex-linked probes and quality control probes used by Illumina. Quality control probes are automatically filtered out in `methylprep` processing with the default `qualityMask`, so there's no need to run `exclude_sex_control_probes` if you processed your data with `methylprep`. Otherwise, we recommend `methylcheck`'s `exclude_sex_control_probes` to remove both sex-linked probes and quality control probes.

[8]:
```
filtered_df = methylcheck.exclude_sex_control_probes(excluded_df, 'epic', no_sex=True,
↪ no_control=True, verbose=True)
```

```
epic: Removed 12360 sex-linked probes from 16 samples. 464871 probes remaining.
```

[9]:
```
filtered_df.head()
```

[9]:

IlmnID	203163220027_R01C01	203163220027_R02C01	203163220027_R03C01	\
cg00000029	0.852	0.749	0.739	
cg00000109	0.943	0.916	0.884	
cg00000165	0.237	0.733	0.374	
cg00000221	0.943	0.937	0.939	
cg00000236	0.922	0.934	0.919	

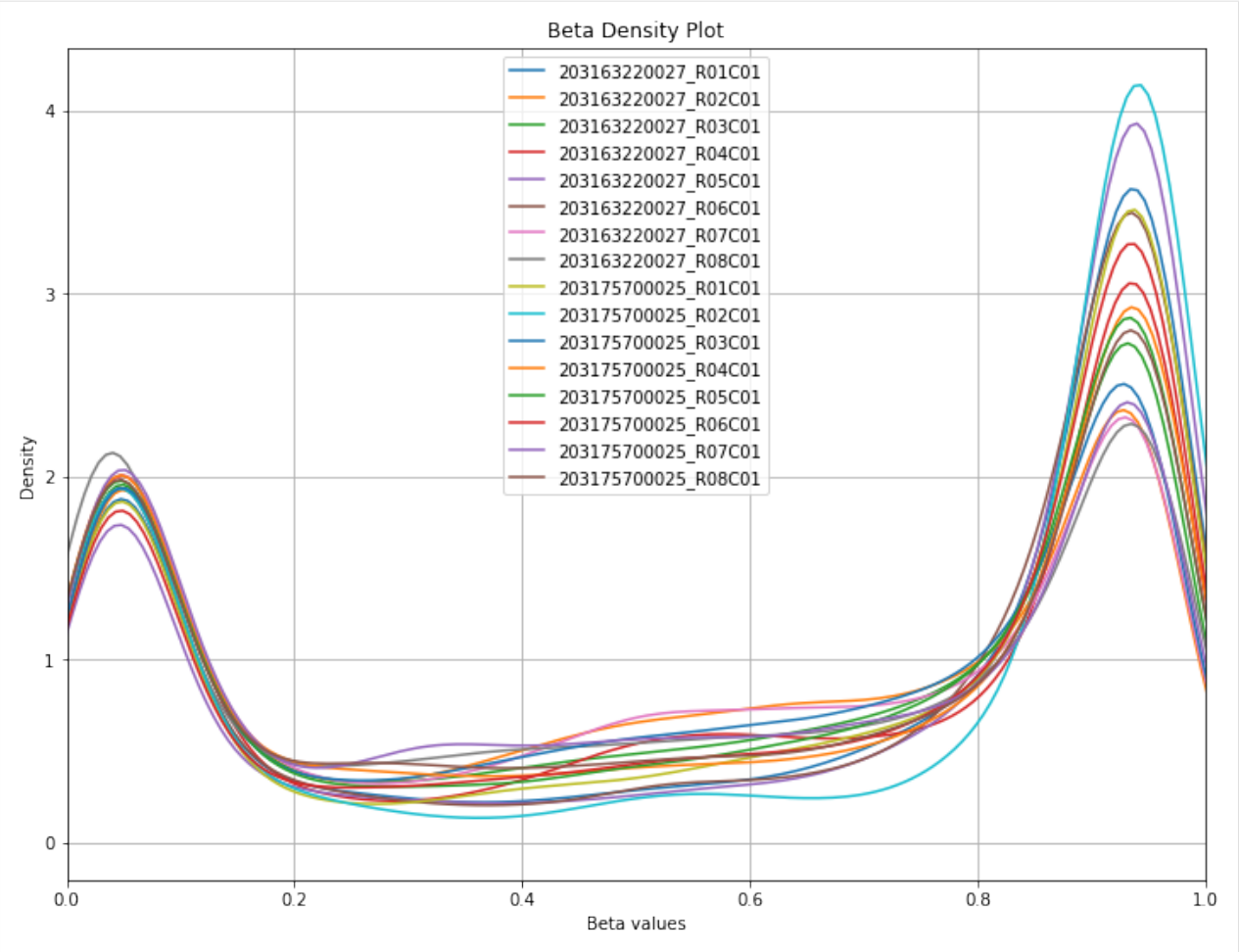
(continues on next page)

(continued from previous page)

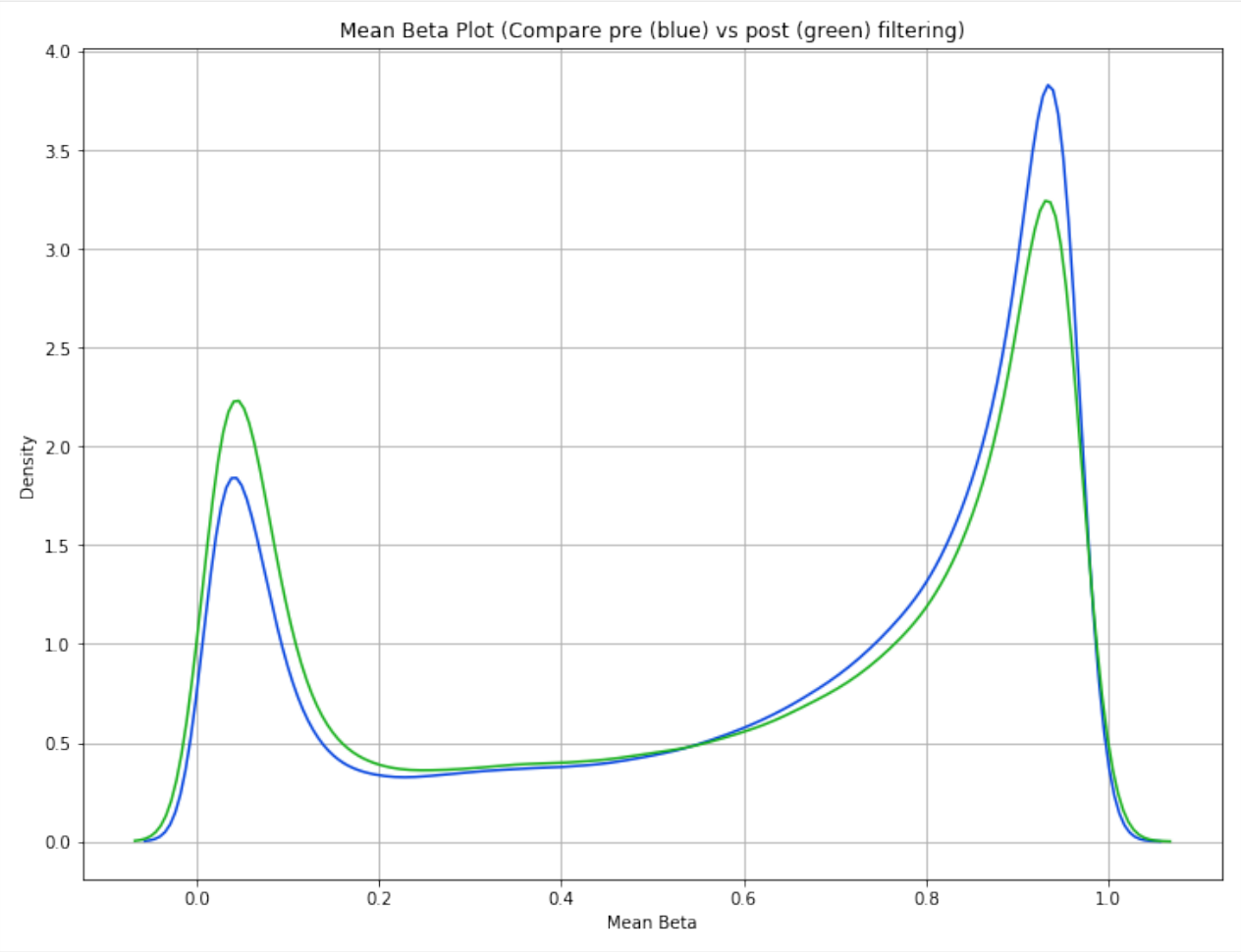
```
            203163220027_R04C01  203163220027_R05C01  203163220027_R06C01  \
IlmnID
cg00000029                  NaN                0.891                0.896
cg00000109                0.951                0.936                0.774
cg00000165                0.639                0.397                0.564
cg00000221                0.926                0.942                0.945
cg00000236                0.881                0.926                0.882

            203163220027_R07C01  203163220027_R08C01  203175700025_R01C01  \
IlmnID
cg00000029                0.757                0.734                0.806
cg00000109                0.932                0.920                0.898
cg00000165                0.486                0.194                0.163
cg00000221                0.938                0.933                0.949
cg00000236                0.927                0.774                0.937

            203175700025_R02C01  203175700025_R03C01  203175700025_R04C01  \
IlmnID
cg00000029                0.881                0.740                0.885
cg00000109                0.938                0.931                0.953
cg00000165                0.557                0.531                0.856
cg00000221                0.936                0.934                0.949
cg00000236                0.957                0.932                0.952

            203175700025_R05C01  203175700025_R06C01  203175700025_R07C01  \
IlmnID
cg00000029                0.693                0.779                0.617
cg00000109                0.899                0.889                0.734
cg00000165                0.171                0.173                0.865
cg00000221                0.944                0.937                0.942
cg00000236                0.923                0.938                0.932

            203175700025_R08C01
IlmnID
cg00000029                0.891
cg00000109                0.892
cg00000165                0.581
cg00000221                0.934
cg00000236                0.951
```

[10]:
```
methylcheck.beta_density_plot(filtered_df)
```

```
WARNING:methylcheck.samples.postprocessQC:Your data contains 5626 missing probe␣
↪values per sample, (90031 overall). For a list per sample, use verbose=True
```



[11]:
```
methylcheck.mean_beta_compare(df, filtered_df)
```



4.3 Quality Control

4.3.1 `controls_report` (a color-coded spreadsheet of control probe performance per sample)

`methylcheck` is geared toward quality control of processed data. To this end, there is a helpful function that summarizes performance of control probes (details on control probes here). To run this function, the control_probes.pkl file output from `methylprep` is required. This report ensures that the chemistry (bisulfite conversion, target specificity, hybridization, staining, etc.) and machine readings are acceptable.

There is an optional portion of this report that relies on values from the poobah file as well. If no poobah file is present, this part is ignored.

Check the guide linked above for more information on how to read these reports. They're intuitively color-coded (green = passing, red = failing, yellow = somewhere in between) so they're easy to read at a glance. There is a colorblind-friendly option included in this function.

We'll walk through examples of the QC pipeline using pre-processed, filtered data (see the Filtering Probes page).

[1]:
```
import methylcheck
from pathlib import Path
filepath = Path('/Users/patriciagirardi/tutorial/GPL21145')
```

[2]:
```
df, metadata = methylcheck.load_both(filepath=filepath)
metadata.head()
```

```
INFO:methylcheck.load_processed:Found several meta_data files; attempting to match␣
↪each with its respective beta_values files in same folders.
WARNING:methylcheck.load_processed:Columns in sample sheet meta data files does not␣
↪match for these files and cannot be combined:['/Users/patriciagirardi/tutorial/
↪GPL21145/GSE147391_GPL21145_meta_data.pkl', '/Users/patriciagirardi/tutorial/
↪GPL21145/sample_sheet_meta_data.pkl']
INFO:methylcheck.load_processed:Multiple meta_data found. Only loading the first file.
INFO:methylcheck.load_processed:Loading 16 samples.
Files: 100%|| 1/1 [00:00<00:00,  5.07it/s]
INFO:methylcheck.load_processed:loaded data (865859, 16) from 1 pickled files (0.223s)
INFO:methylcheck.load_processed:meta.Sample_IDs match data.index (OK)
```

[2]:

	GSM_ID	Sample_Name	Sentrix_ID	Sentrix_Position	\
2	GSM4429898	Grade II rep3	203163220027	R01C01	
3	GSM4429899	Grade III rep1	203163220027	R02C01	
12	GSM4429908	Grade IV rep5	203163220027	R03C01	
15	GSM4429911	Grade IV rep8	203163220027	R04C01	
6	GSM4429902	Grade II rep6	203163220027	R05C01	

	source	histological diagnosis	description	\
2	Resected glioma	Diffuse astrocytoma (II)	Glioma	
3	Resected glioma	Anaplastic astrocytoma (III)	Glioma	
12	Resected glioma	Glioblastoma (IV)	Glioma	
15	Resected glioma	Glioblastoma (IV)	Glioma	
6	Resected glioma	Oligodendroglioma (II)	Glioma	

	Sample_ID
2	203163220027_R01C01
3	203163220027_R02C01
12	203163220027_R03C01
15	203163220027_R04C01
6	203163220027_R05C01

[3]:
```
methylcheck.controls_report(filepath=filepath)
```

```
INFO:methylprep.files.manifests:Reading manifest file: MethylationEPIC_v-1-0_B4.
↪CoreColumns.csv
```

```
203175700025_R01C01 GA r=0.81 ±0.23 p<0.0 |CT r=0.82 ±0.15 p<0.0
203175700025_R02C01 GA r=0.9 ±0.16 p<0.0 |CT r=0.87 ±0.13 p<0.0
203163220027_R01C01 GA r=0.86 ±0.19 p<0.0 |CT r=0.83 ±0.14 p<0.0
203163220027_R02C01 GA r=0.88 ±0.16 p<0.0 |CT r=0.85 ±0.13 p<0.0
203175700025_R05C01 GA r=0.87 ±0.14 p<0.0 |CT r=0.84 ±0.11 p<0.0
203175700025_R06C01 GA r=0.91 ±0.12 p<0.0 |CT r=0.87 ±0.1 p<0.0
203163220027_R05C01 GA r=0.87 ±0.16 p<0.0 |CT r=0.85 ±0.12 p<0.0
203163220027_R06C01 GA r=0.9 ±0.14 p<0.0 |CT r=0.85 ±0.12 p<0.0
203175700025_R03C01 GA r=0.85 ±0.17 p<0.0 |CT r=0.85 ±0.13 p<0.0
203175700025_R04C01 GA r=0.87 ±0.15 p<0.0 |CT r=0.87 ±0.11 p<0.0
203175700025_R07C01 GA r=0.88 ±0.13 p<0.0 |CT r=0.85 ±0.11 p<0.0
203175700025_R08C01 GA r=0.9 ±0.13 p<0.0 |CT r=0.87 ±0.1 p<0.0
203163220027_R03C01 GA r=0.84 ±0.17 p<0.0 |CT r=0.83 ±0.13 p<0.0
```

(continues on next page)

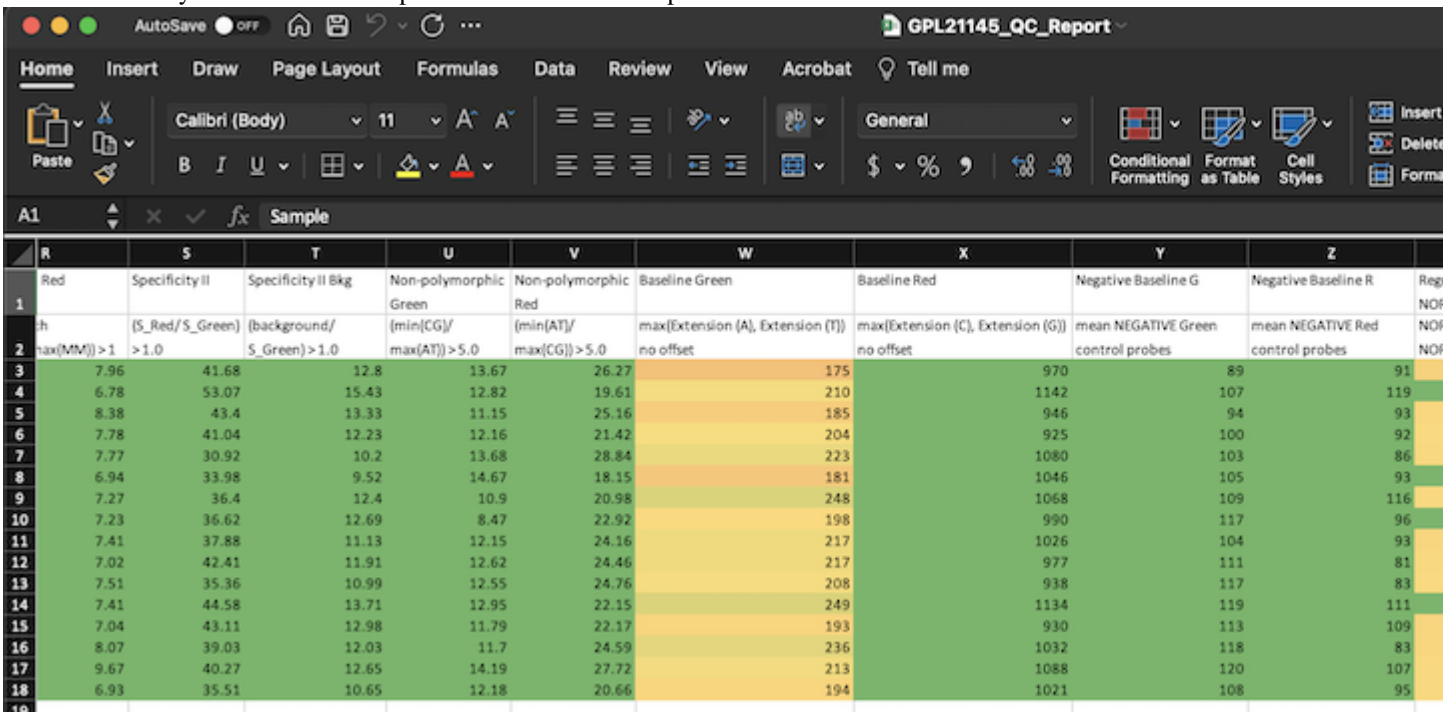
(continued from previous page)

```
203163220027_R07C01 GA r=0.85 ±0.16 p<0.0 |CT r=0.86 ±0.11 p<0.0
203163220027_R08C01 GA r=0.87 ±0.15 p<0.0 |CT r=0.85 ±0.12 p<0.0
```

```
INFO:methylcheck.reports.controls_report:Predicting Sex...
INFO:methylprep.files.manifests:Reading manifest file: MethylationEPIC_v-1-0_B4.
↪CoreColumns.csv
```

```
203163220027_R04C01 GA r=0.87 ±0.16 p<0.0 |CT r=0.87 ±0.12 p<0.0
(865859, 16) (865859, 16)
```

The color-coded results are contained in an excel file that will be saved in the same directory that was specified as an input. It will look similar to this:



	R	S	T	U	V	W	X	Y	Z
1	Red	Specificity II	Specificity II Bkg	Non-polymorphic Green	Non-polymorphic Red	Baseline Green	Baseline Red	Negative Baseline G	Negative Baseline R
2	h max(MM)) > 1	(S_Red/S_Green) > 1.0	(background/ S_Green) > 1.0	(min(CG)/ max(AT)) > 5.0	(min(AT)/ max(CG)) > 5.0	max(Extension (A), Extension (T)) no offset	max(Extension (C), Extension (G)) no offset	mean NEGATIVE Green control probes	mean NEGATIVE Red control probes
3	7.96	41.68	12.8	13.67	26.27	175	970	89	91
4	6.78	53.07	15.43	12.82	19.61	210	1142	107	119
5	8.38	43.4	13.33	11.15	25.16	185	946	94	93
6	7.78	41.04	12.23	12.16	21.42	204	925	100	92
7	7.77	30.92	10.2	13.68	28.84	223	1080	103	86
8	6.94	33.98	9.52	14.67	18.15	181	1046	105	93
9	7.27	36.4	12.4	10.9	20.98	248	1068	109	116
10	7.23	36.62	12.69	8.47	22.92	198	990	117	96
11	7.41	37.88	11.13	12.15	24.16	217	1026	104	93
12	7.02	42.41	11.91	12.62	24.46	217	977	111	81
13	7.51	35.36	10.99	12.55	24.76	208	938	117	83
14	7.41	44.58	13.71	12.95	22.15	249	1134	119	111
15	7.04	43.11	12.98	11.79	22.17	193	930	113	109
16	8.07	39.03	12.03	11.7	24.59	236	1032	118	83
17	9.67	40.27	12.65	14.19	27.72	213	1088	120	107
18	6.93	35.51	10.65	12.18	20.66	194	1021	108	95

The "Passing Probes" column is the column that relies on values from the poobah file. This is a measure of how many probes failed in each sample (detection p-value > 0.05). The p-value cut off is adjustable with the `pval_sig` argument, which is set to 0.05 by default.

Notice the final column ("Result") where most samples are passing. This column is calculated by checking that all of the QC columns are above a minimum threshold. This threshold is adjustable with the `passing` argument (set to 0.7 by default). - If the poobah file is included and 20% or more of probes fail, the Result is automatically FAIL. - If 70% of the columns are passing, the result is "OK" or passing. - If more than 70% are passing, but less than 100%, the "OK" will have a number next to it to specify what percentage of columns passed. - If less than 70% of the columns passed, the result is either FAIL or MARGINAL (based on how close to the 70% threshold it got).

The predicted sex column is based on the median values of methylation measurements on the X and Y chromosomes. If `yMed - xMed` is less than whatever the specified cutoff value is, it is predicted female. Otherwise, the predicted sex is male. Also, for samples from female subjects, at least 90% of the Y chromosome probes should fail. If there is a sample sheet that includes "sex" or "gender", the reporter will also flag any mismatches between the predicted sex and the specified sex (this does not affect the Result column).

4.3.2 Quality Control in the IDE

If you want to run a quick quality control check within your CLI or IDE (with no output files saved), use the `run_qc` function. This is a simplified version of the full `controls_report` function.

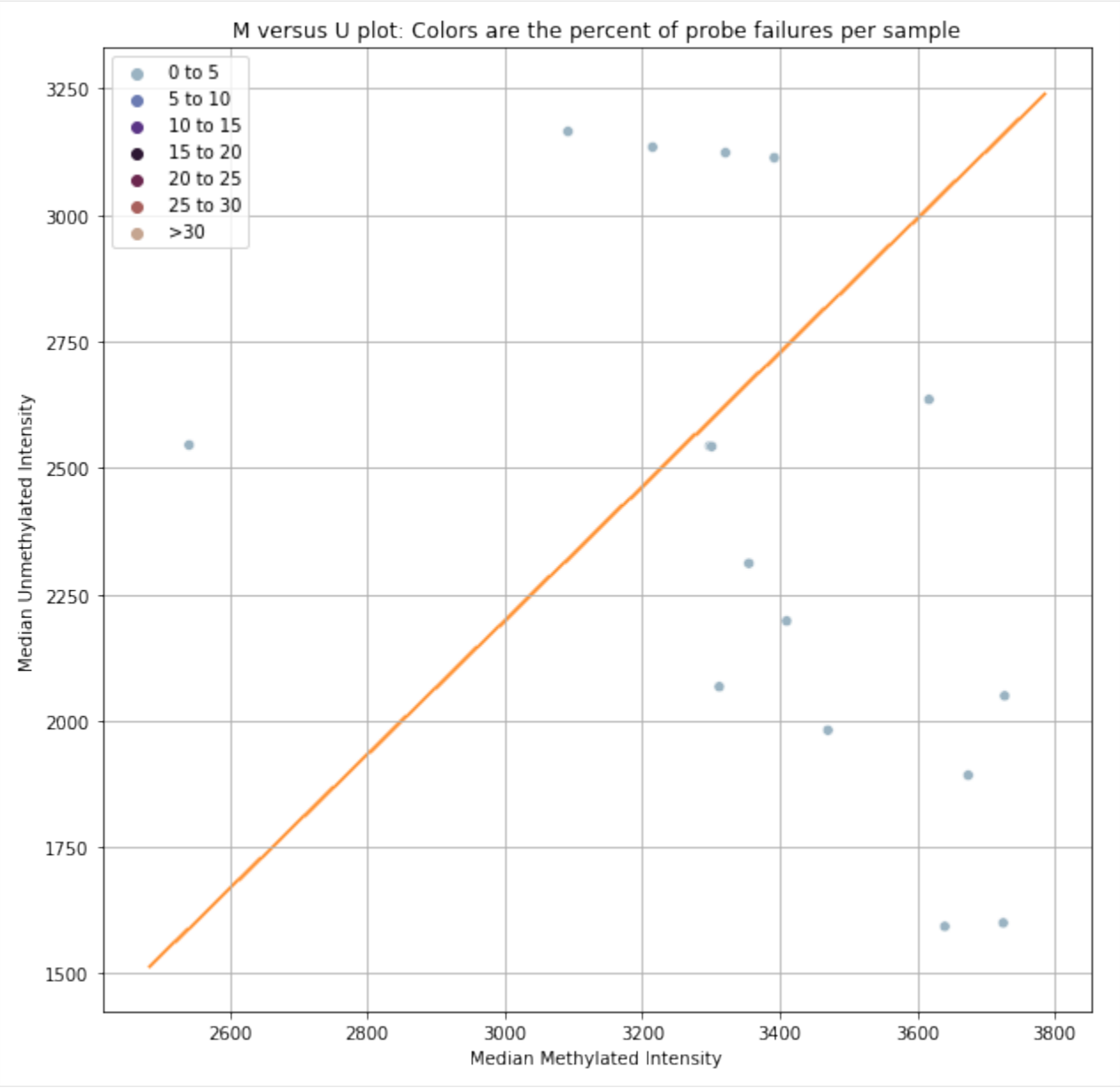
The first and second plots shown below are based on the `qc_signal_intensity` function, which suggests sample outliers based on methylated and unmethylated signal intensity. The cutoff value is based on `minfi`'s chosen cutoff value. If a sample falls below the dotted line in the second chart, it is potentially poor quality (due to low fluorescence). This is a very specific way of identifying poor quality samples and we recommend taking a more holistic approach (using `controls_report()`). However, this can still be useful information in some circumstances, which is why we've included it in the original qc pipeline.

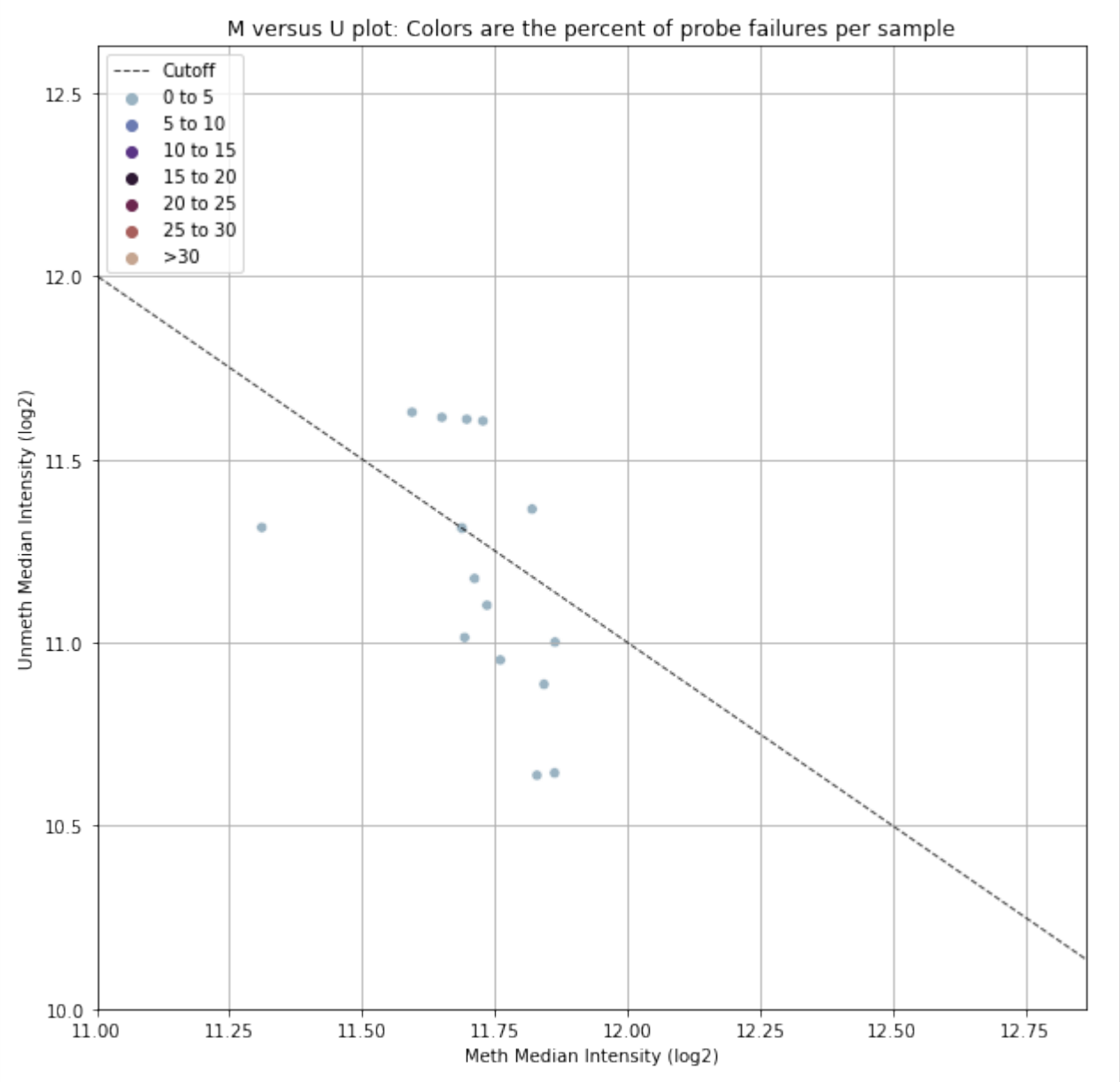
The next set of plots cover hybridization, staining, bisulfite conversion, specificity, target removal, extension, negative controls, and non-polymorphic binding, but can be more difficult to interpret in this graph format. For more, consult Illumina's technical document on what the expected values for these graphs should be.

The final set of plots is beta distributions by probe type (Type I and Type II); the distributions of Type I probes are split by color channel as well.

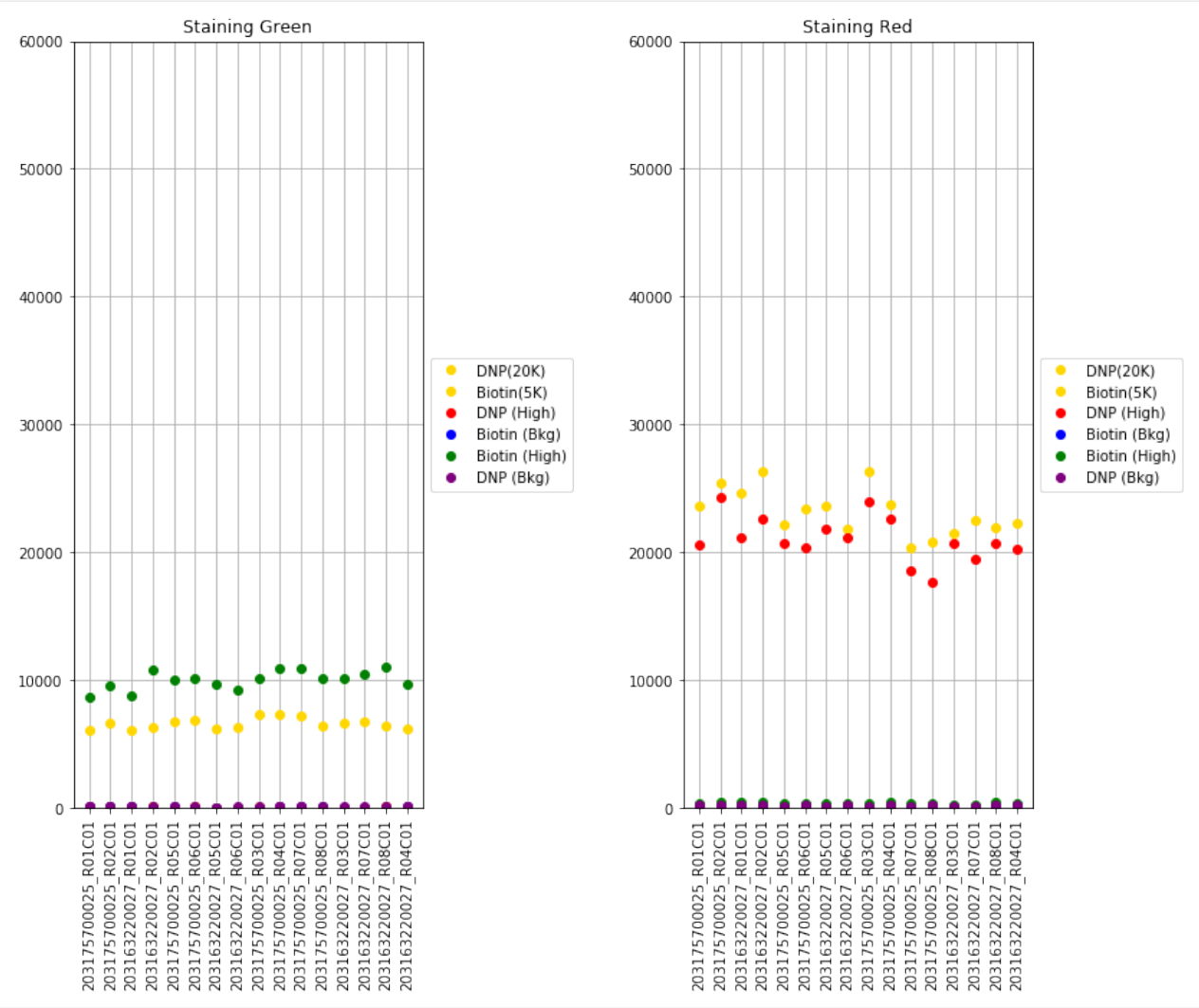
This function is to be used before any removal of probes or samples. If you remove probes or samples from your beta value dataframe, this function will no longer run due to inconsitant rows and/or columns in other dataframes (such as m-value).

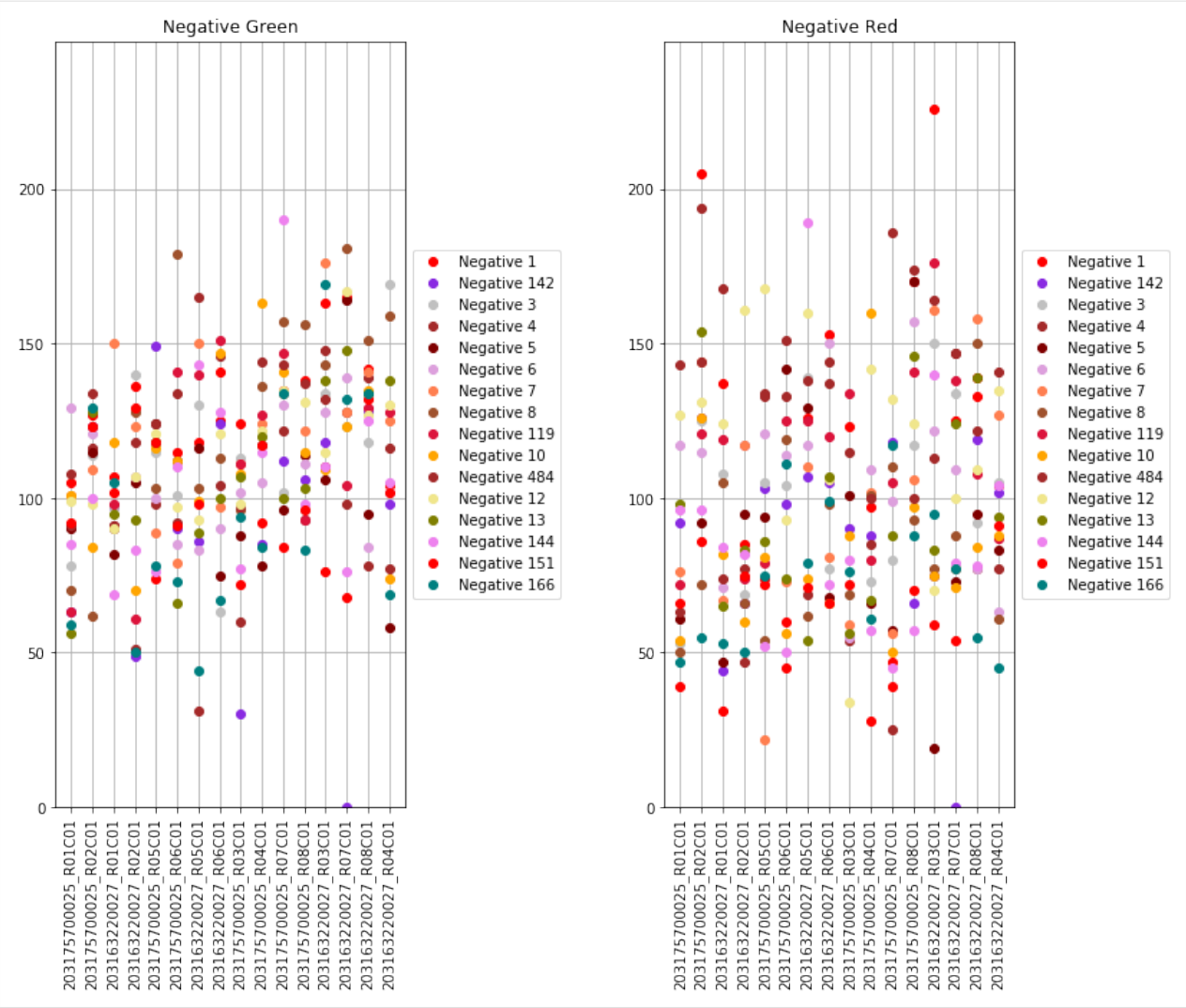
[4]:
```
methylcheck.run_qc(filepath)
```

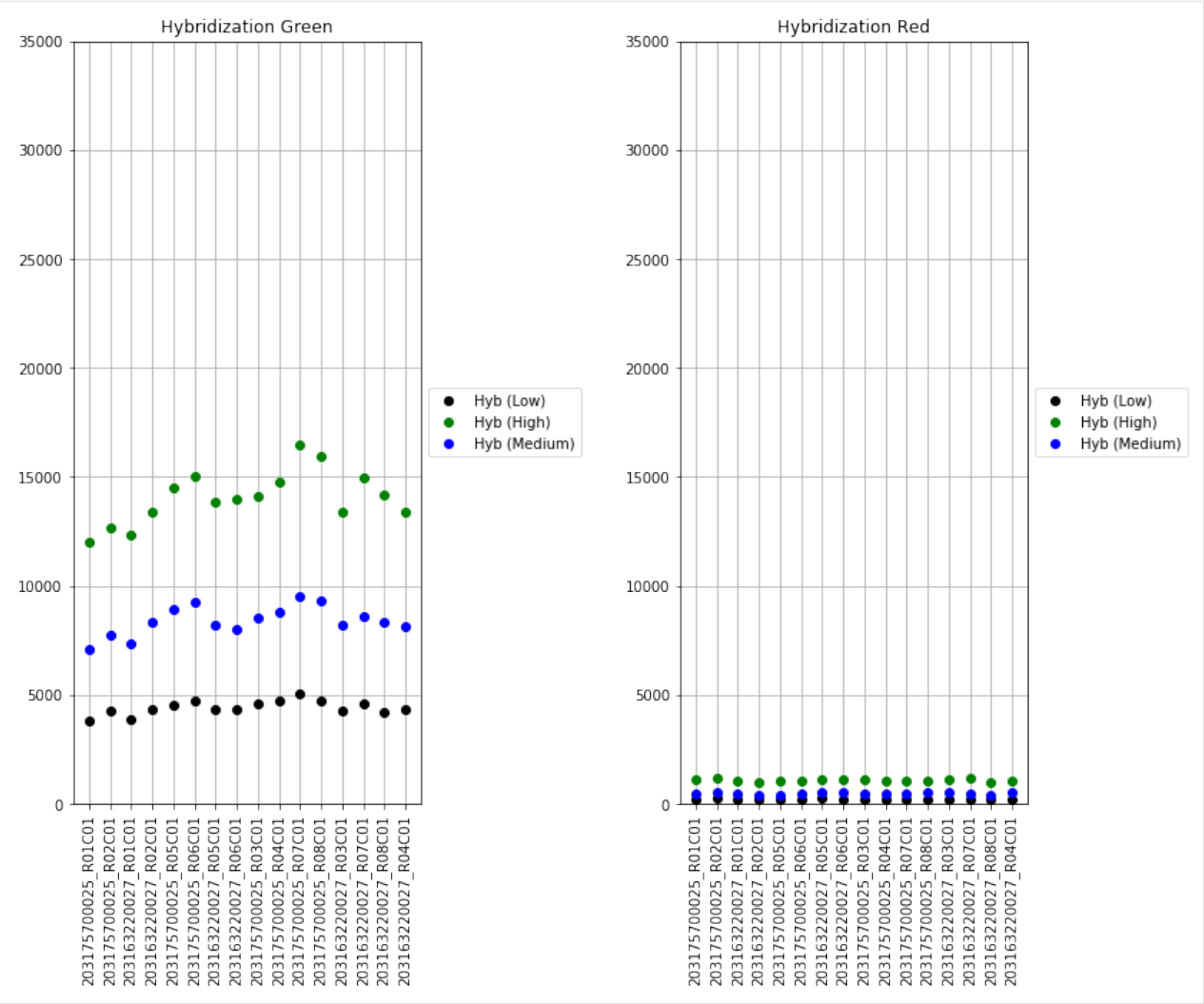

M versus U plot: Colors are the percent of probe failures per sample
0 to 5
5 to 10
10 to 15
15 to 20
20 to 25
25 to 30
>30
Median Unmethylated Intensity
Median Methylated Intensity
3250
3000
2750
2500
2250
2000
1750
1500
2600
2800
3000
3200
3400
3600
3800

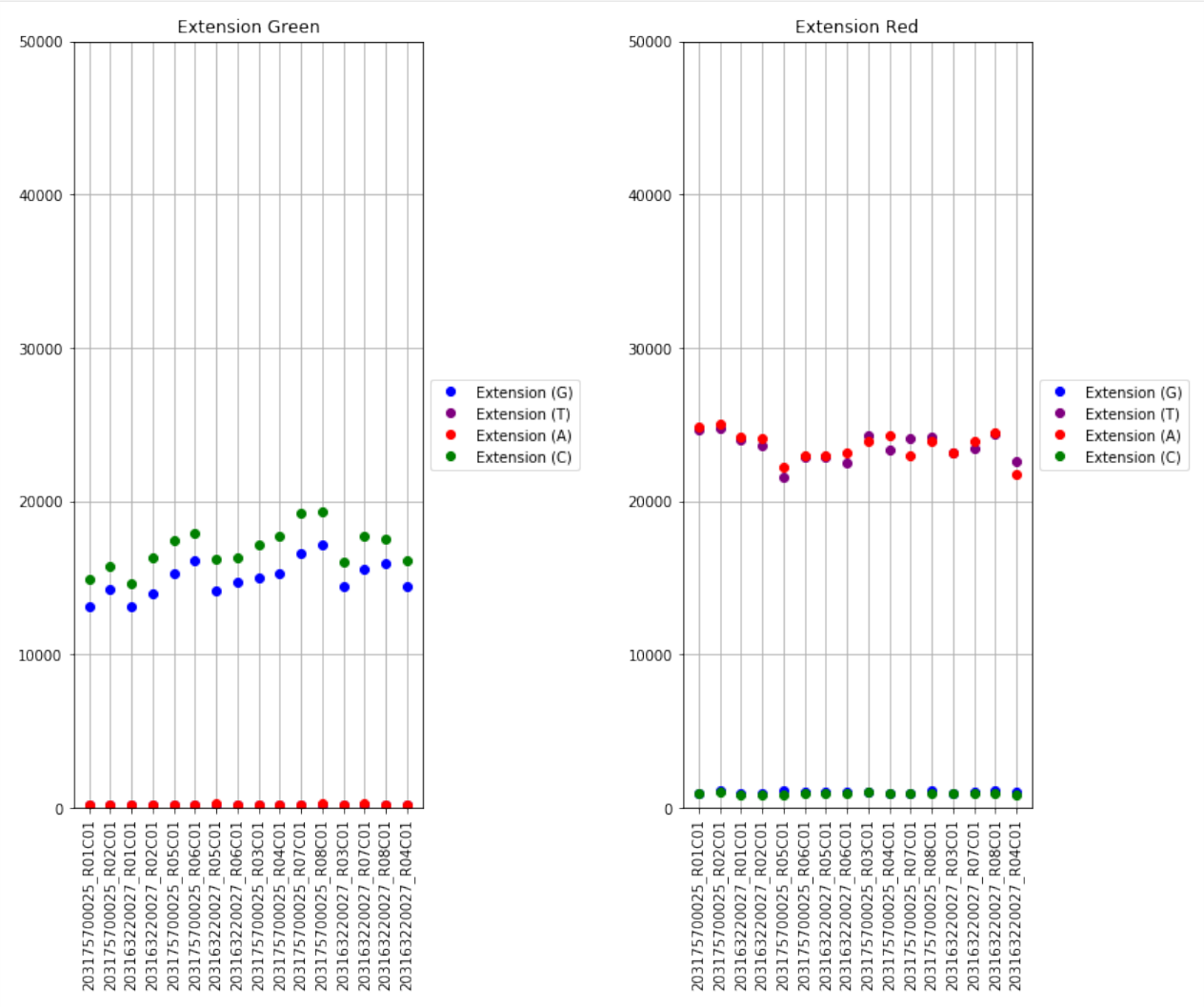


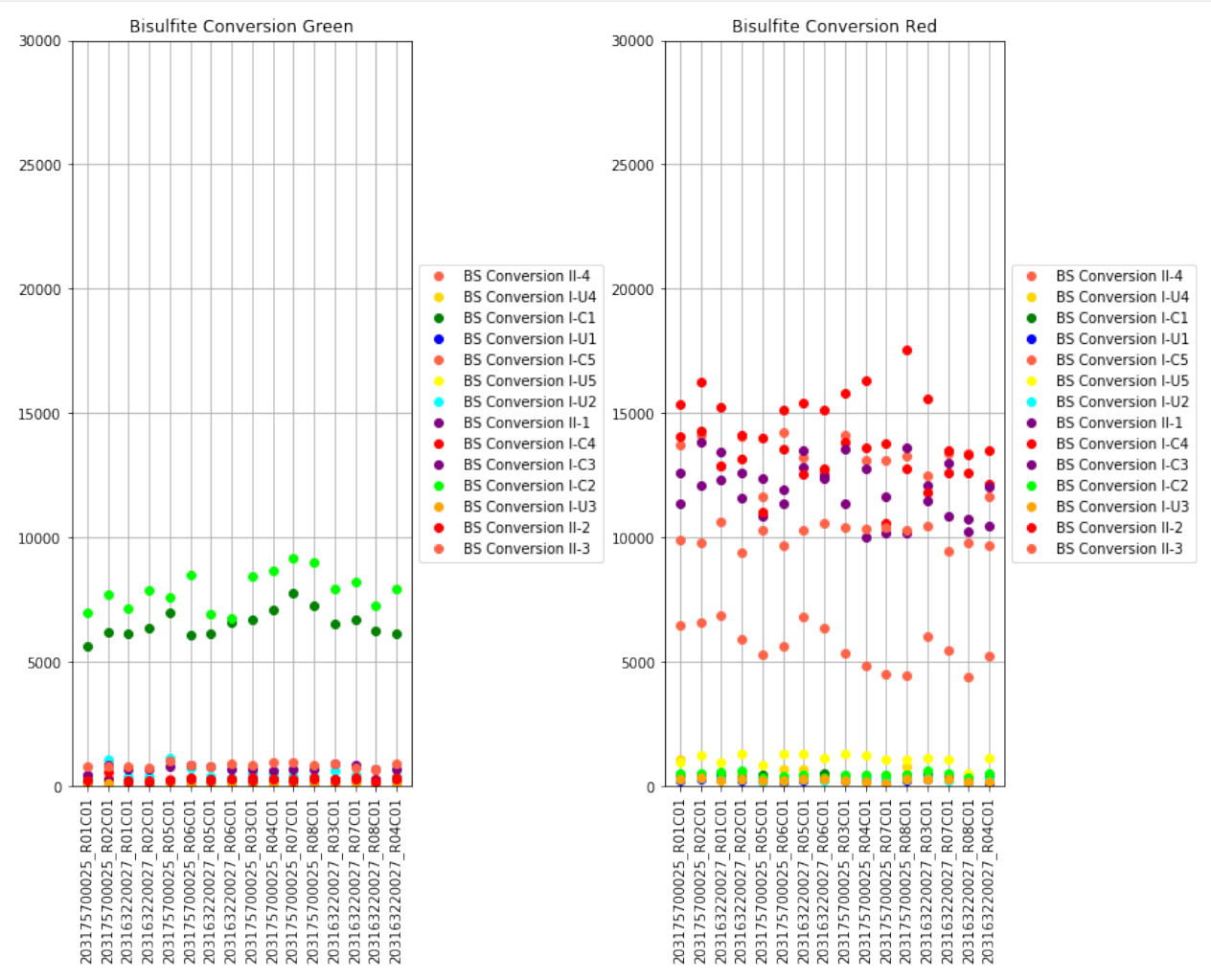

```
List of Bad Samples
['203163220027_R01C01', '203163220027_R03C01', '203163220027_R04C01', '203163220027_
↪R05C01', '203163220027_R06C01', '203163220027_R08C01', '203175700025_R01C01',
↪'203175700025_R02C01', '203175700025_R05C01', '203175700025_R06C01']
```

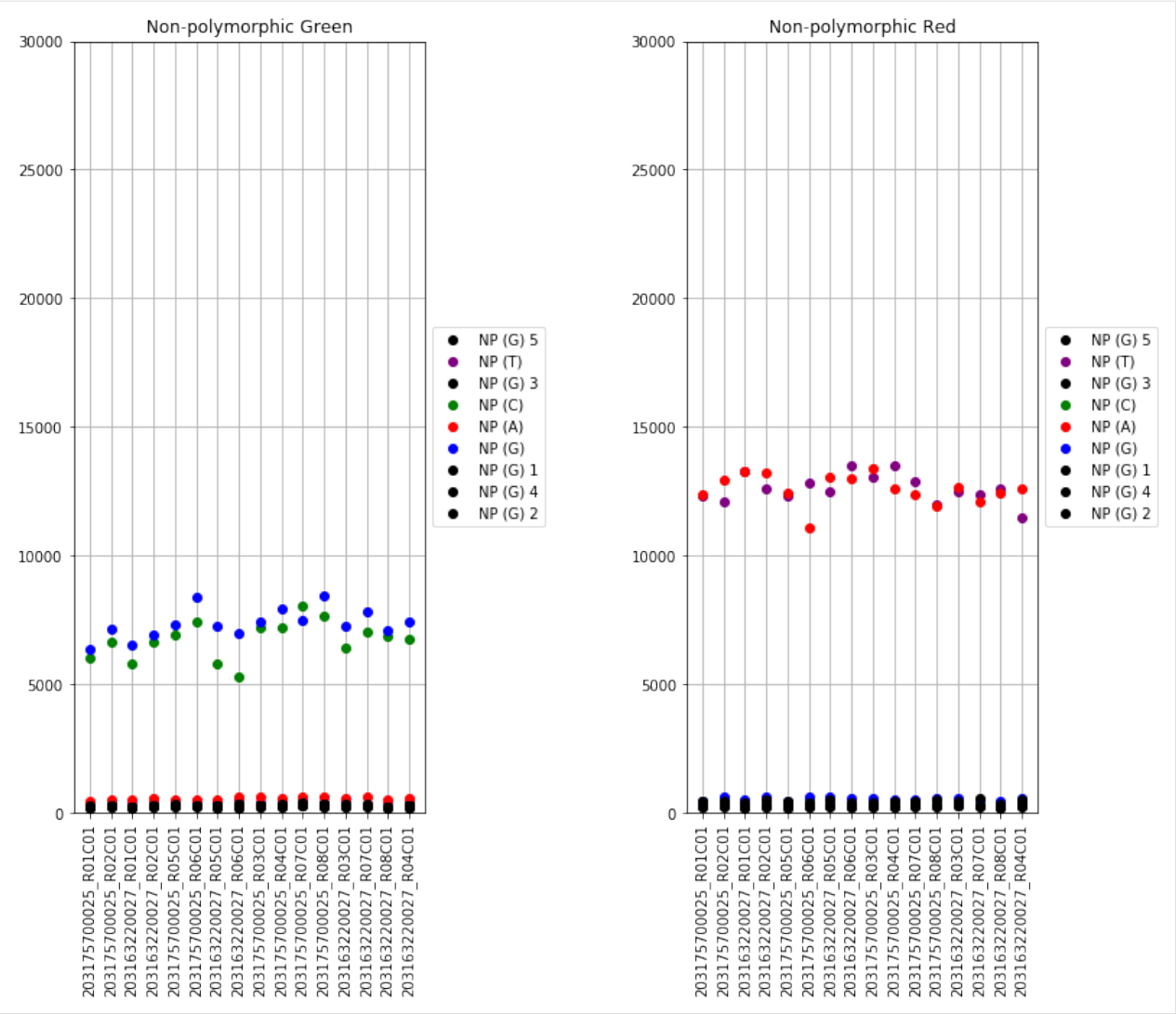

Staining Green
Staining Red
60000
50000
40000
30000
20000
10000
0
DNP(20K)
Biotin(5K)
DNP (High)
Biotin (Bkg)
Biotin (High)
DNP (Bkg)
203175700025_R01C01
203175700025_R02C01
203163220027_R01C01
203163220027_R02C01
203175700025_R05C01
203175700025_R06C01
203163220027_R05C01
203163220027_R06C01
203175700025_R03C01
203175700025_R04C01
203175700025_R07C01
203175700025_R08C01
203163220027_R03C01
203163220027_R07C01
203163220027_R08C01
203163220027_R04C01

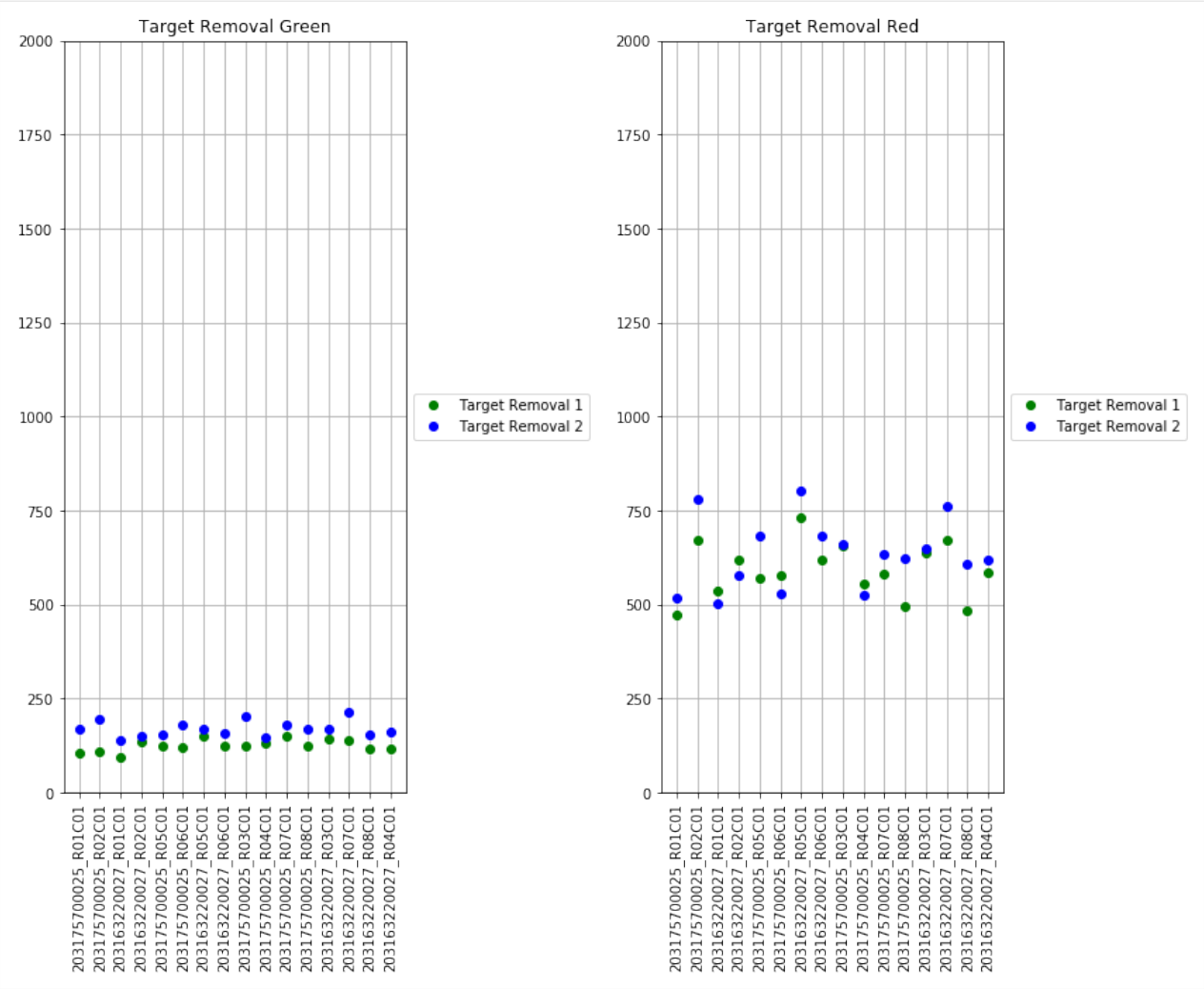

Negative Green
Negative Red
200
150
100
50
0
Negative 1
Negative 142
Negative 3
Negative 4
Negative 5
Negative 6
Negative 7
Negative 8
Negative 119
Negative 10
Negative 484
Negative 12
Negative 13
Negative 144
Negative 151
Negative 166
203175700025_R01C01
203175700025_R02C01
203163220027_R01C01
203163220027_R02C01
203175700025_R05C01
203175700025_R06C01
203163220027_R05C01
203163220027_R06C01
203175700025_R03C01
203175700025_R04C01
203175700025_R07C01
203175700025_R08C01
203163220027_R03C01
203163220027_R07C01
203163220027_R08C01
203163220027_R04C01

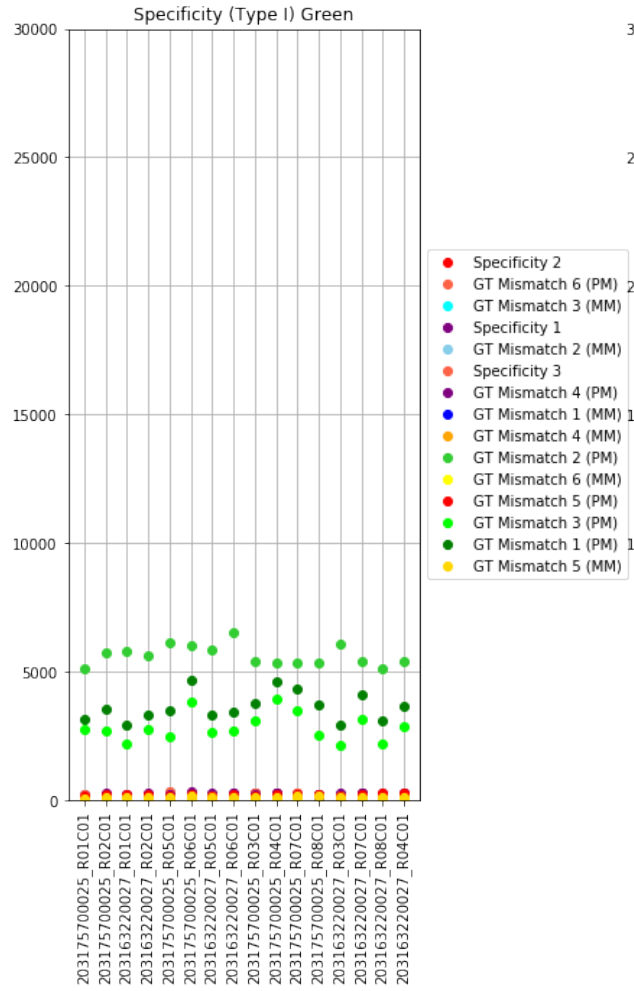

Hybridization Green
Hybridization Red
Hyb (Low)
Hyb (High)
Hyb (Medium)
0
5000
10000
15000
20000
25000
30000
35000
203175700025_R01C01
203175700025_R02C01
203163220027_R01C01
203163220027_R02C01
203175700025_R05C01
203175700025_R06C01
203163220027_R05C01
203163220027_R06C01
203175700025_R03C01
203175700025_R04C01
203175700025_R07C01
203175700025_R08C01
203163220027_R03C01
203163220027_R07C01
203163220027_R08C01
203163220027_R04C01

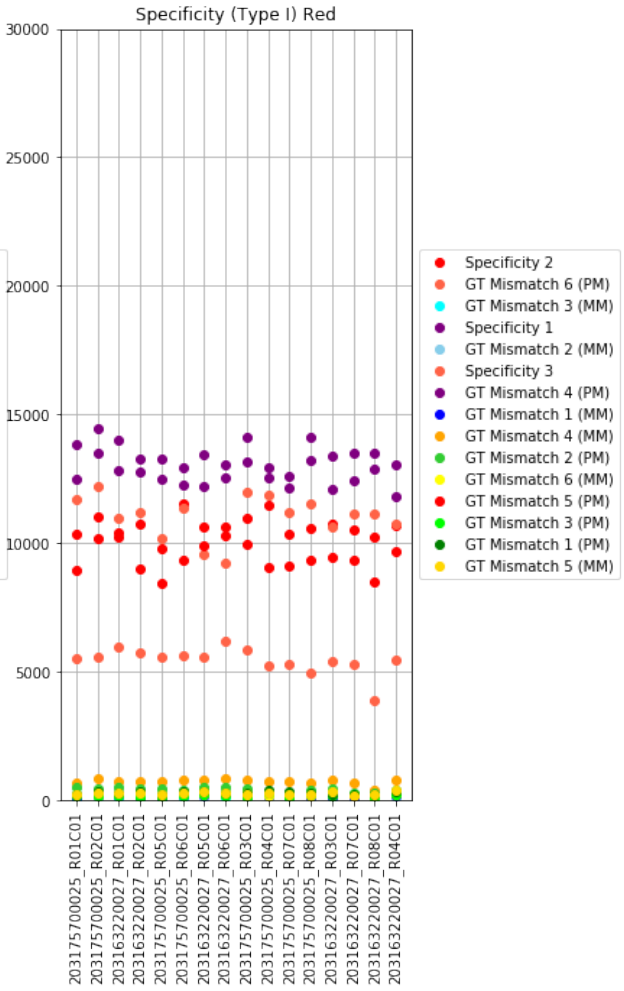

Extension Green
50000
40000
30000
20000
10000
0
Extension (G)
Extension (T)
Extension (A)
Extension (C)
Extension Red
50000
40000
30000
20000
10000
0
Extension (G)
Extension (T)
Extension (A)
Extension (C)


Bisulfite Conversion Green
Bisulfite Conversion Red
30000
25000
20000
15000
10000
5000
0
BS Conversion II-4
BS Conversion I-U4
BS Conversion I-C1
BS Conversion I-U1
BS Conversion I-C5
BS Conversion I-U5
BS Conversion I-U2
BS Conversion II-1
BS Conversion I-C4
BS Conversion I-C3
BS Conversion I-C2
BS Conversion I-U3
BS Conversion II-2
BS Conversion II-3
203175700025_R01C01
203175700025_R02C01
203163220027_R01C01
203163220027_R02C01
203175700025_R05C01
203175700025_R06C01
203163220027_R05C01
203163220027_R06C01
203175700025_R03C01
203175700025_R04C01
203175700025_R07C01
203175700025_R08C01
203163220027_R03C01
203163220027_R07C01
203163220027_R08C01
203163220027_R04C01


Non-polymorphic Green
Non-polymorphic Red
30000
25000
20000
15000
10000
5000
0
NP (G) 5
NP (T)
NP (G) 3
NP (C)
NP (A)
NP (G)
NP (G) 1
NP (G) 4
NP (G) 2
203175700025_R01C01
203175700025_R02C01
203163220027_R01C01
203163220027_R02C01
203175700025_R05C01
203175700025_R06C01
203163220027_R05C01
203163220027_R06C01
203175700025_R03C01
203175700025_R04C01
203175700025_R07C01
203175700025_R08C01
203163220027_R03C01
203163220027_R07C01
203163220027_R08C01
203163220027_R04C01

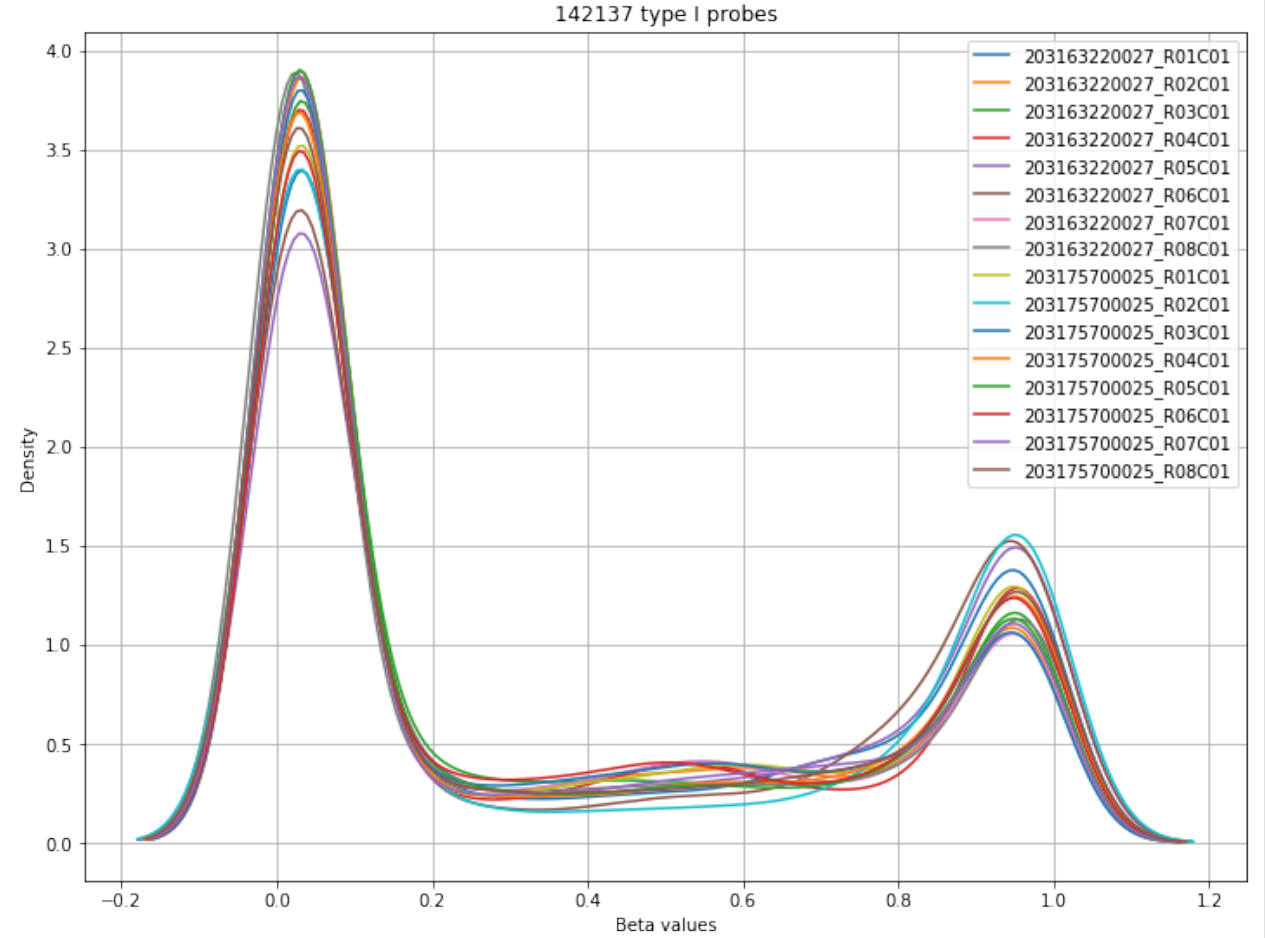

Target Removal Green
Target Removal Red
Target Removal 1
Target Removal 2
2000
1750
1500
1250
1000
750
500
250
0
203175700025_R01C01
203175700025_R02C01
203163220027_R01C01
203163220027_R02C01
203175700025_R05C01
203175700025_R06C01
203163220027_R05C01
203163220027_R06C01
203175700025_R03C01
203175700025_R04C01
203175700025_R07C01
203175700025_R08C01
203163220027_R03C01
203163220027_R07C01
203163220027_R08C01
203163220027_R04C01





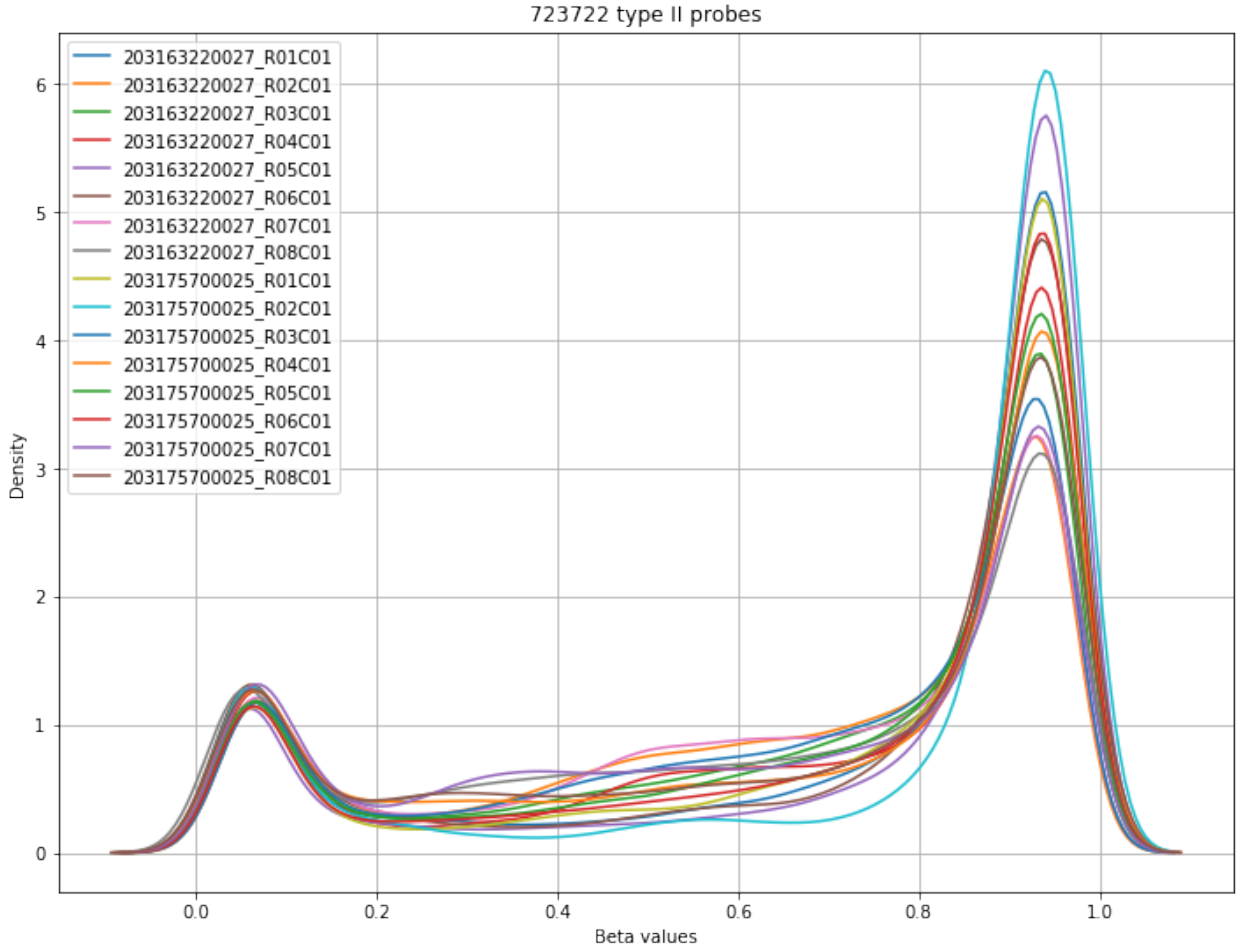

```
INFO:methylprep.files.manifests:Reading manifest file: MethylationEPIC_v-1-0_B4.
↪CoreColumns.csv
WARNING:methylcheck.samples.postprocessQC:Your data contains 21853 missing probe␣
↪values per sample, (349658 overall). For a list per sample, use verbose=True
```

```
Found 142137 type I probes.
```



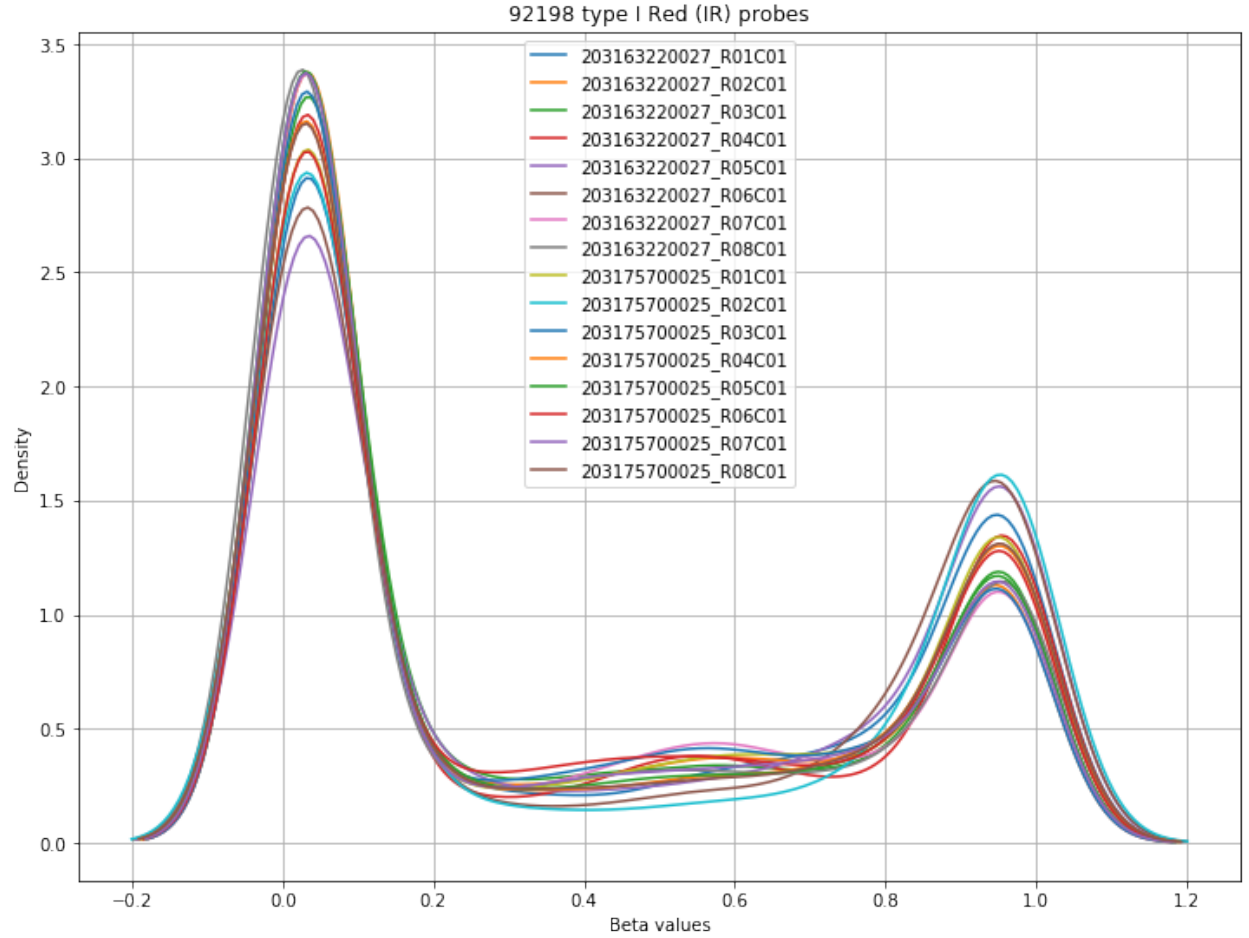

```
WARNING:methylcheck.samples.postprocessQC:Your data contains 89427 missing probe␣
↪values per sample, (1430832 overall). For a list per sample, use verbose=True
```

```
Found 723722 type II probes.
```



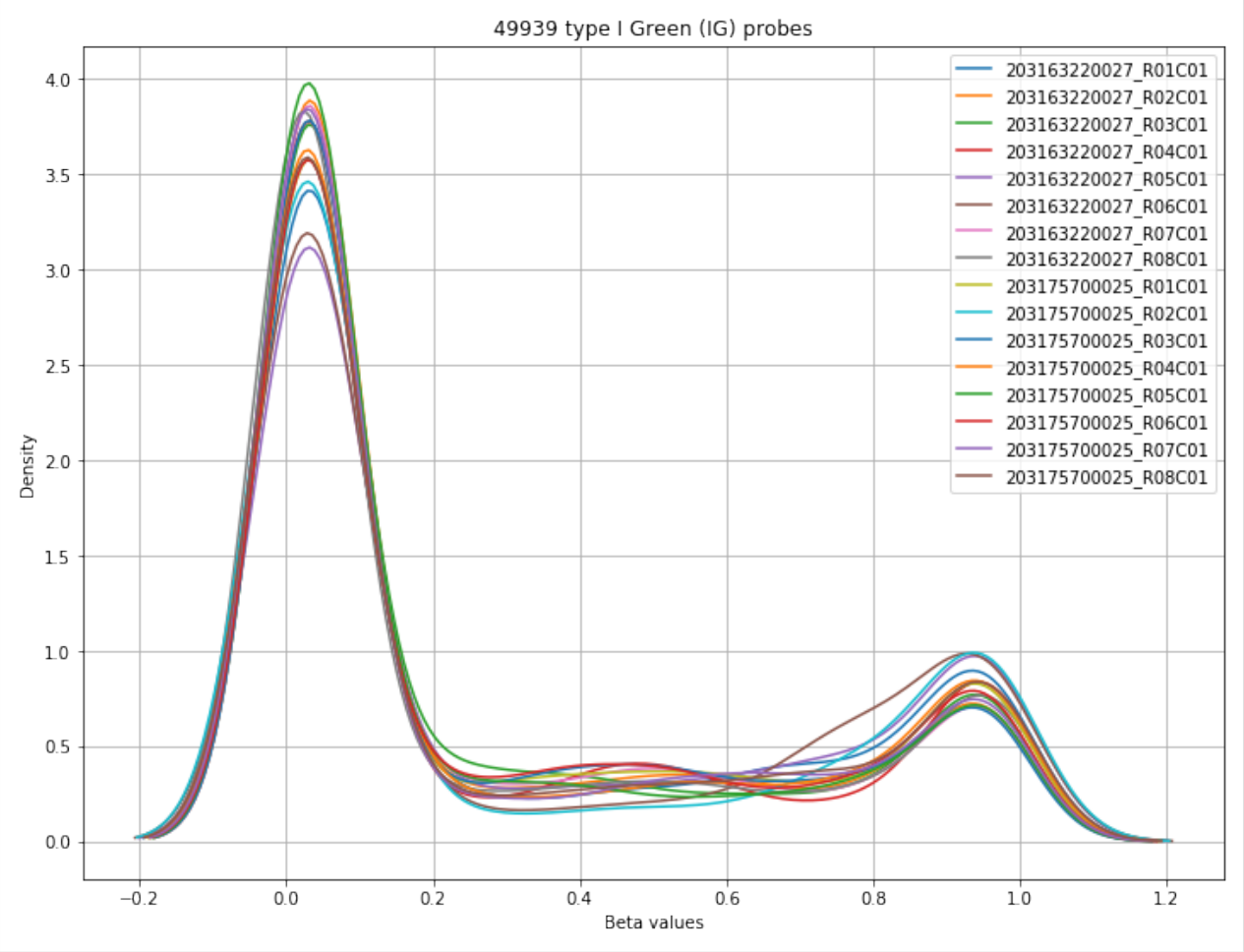

```
WARNING:methylcheck.samples.postprocessQC:Your data contains 15076 missing probe␣
↪values per sample, (241225 overall). For a list per sample, use verbose=True
```

```
Found 92198 type I Red (IR) probes.
```




```
WARNING:methylcheck.samples.postprocessQC:Your data contains 6777 missing probe␣
↪values per sample, (108433 overall). For a list per sample, use verbose=True
```

```
Found 49939 type I Green (IG) probes.
```

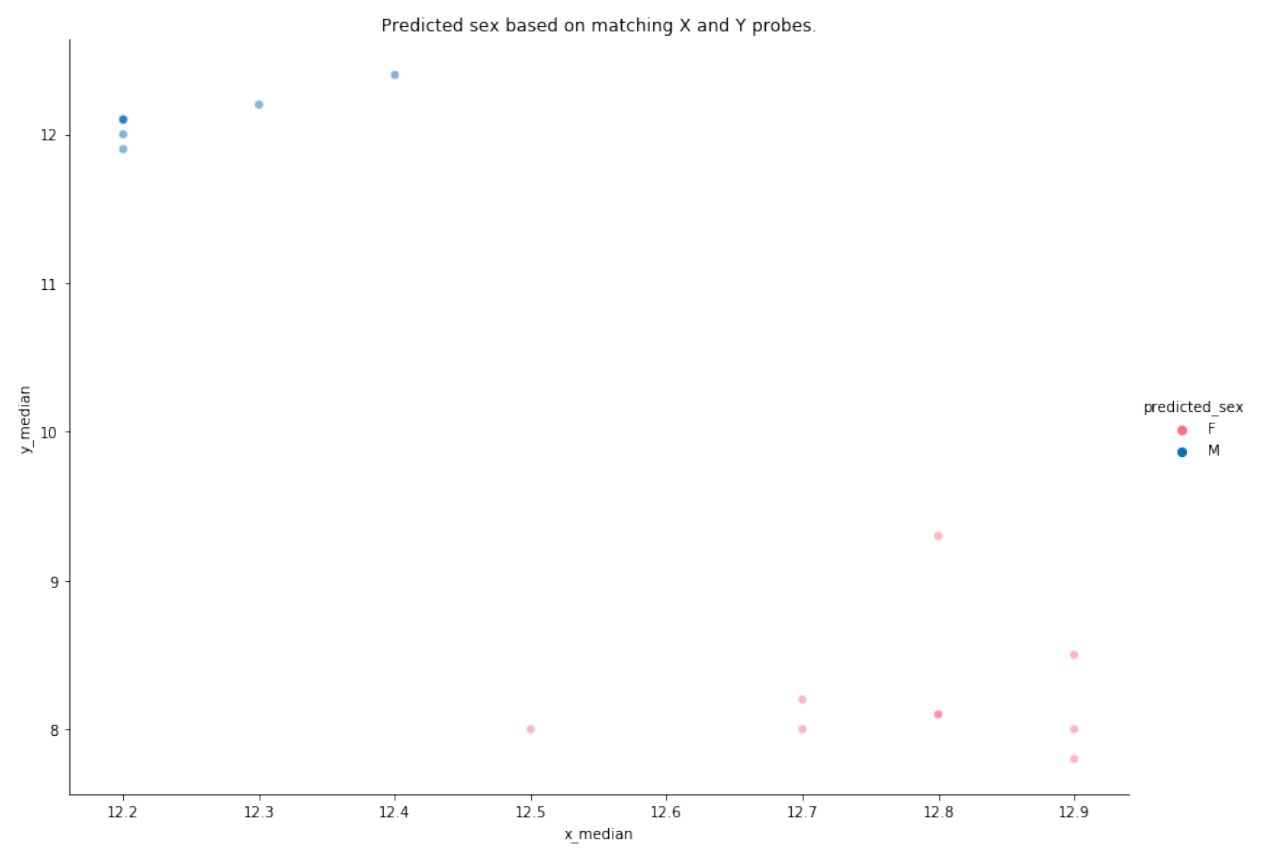


4.3.3 Predicting Sex

If you want to predict the sex of your samples without generating the entire QC report, you need the methylated and unmethylated .pkl files generated by `methylprep` and the `get_sex` function. See below for an example of its usage.

[5]:
```
import methylcheck
from pathlib import Path
filepath = Path('/Users/patriciagirardi/tutorial/GPL21145')
(meth,unmeth) = methylcheck.load(filepath, format='meth_df')

methylcheck.get_sex((meth, unmeth), plot=True)
```

```
100%|| 16/16 [00:00<00:00, 491.79it/s]
100%|| 16/16 [00:00<00:00, 1415.80it/s]
INFO:methylcheck.load_processed:(865859, 16) (865859, 16)
INFO:methylprep.files.manifests:Reading manifest file: MethylationEPIC_v-1-0_B4.
↪CoreColumns.csv
WARNING:methylcheck.predict.sex:sample_failure_percent index did not align with␣
↪output data index
```



[5]:

	x_median	y_median	predicted_sex
203163220027_R01C01	12.7	8.0	F
203163220027_R02C01	12.3	12.2	M
203163220027_R03C01	12.8	9.3	F
203163220027_R04C01	12.8	8.1	F
203163220027_R05C01	12.2	12.0	M
203163220027_R06C01	12.2	12.1	M
203163220027_R07C01	12.9	8.5	F
203163220027_R08C01	12.7	8.2	F
203175700025_R01C01	12.5	8.0	F
203175700025_R02C01	12.2	12.1	M
203175700025_R03C01	12.9	8.0	F
203175700025_R04C01	12.9	7.8	F
203175700025_R05C01	12.8	8.1	F
203175700025_R06C01	12.2	11.9	M
203175700025_R07C01	12.4	12.4	M
203175700025_R08C01	12.2	12.1	M

Darker, smaller markers indicate less variability and a higher confidence in the predicted sex. Our data is pretty clean and has a good separation between the two sexes. It is occasionally the case that some samples will be misclassified. Those warrant further investigation as they are often poor quality samples that should be excluded.

4.3.4 Report PDF tool

This is a fully customizable class template for making more advanced quality control reports. It will call a batch of plotting functions and compile a PDF with annotation and save file to disk.

- kwargs: `filename, poobah_max_percent, pval_cutoff, title, author, subject, keywords, outpath, path`

If you want a quick one-line version with no further report editing, try `methylcheck.ReportPDF(runme=True)`. Otherwise, these are the 3 steps for using the ReportPDF class:

[8]:
```
my_report = methylcheck.ReportPDF(filename='methylcheck_report.pdf',
                                  title="Example Report",
                                  path=filepath)
```

[9]:
```
my_report.run_qc()
```

```
INFO:methylcheck.reports.qc_report:Data loaded
INFO:methylcheck.reports.qc_report:Beta Density Plot
```

```
['203163220027_R01C01', '203163220027_R02C01', '203163220027_R03C01', '203163220027_
↪R05C01', '203163220027_R07C01', '203163220027_R08C01', '203175700025_R01C01',
↪'203175700025_R03C01', '203175700025_R04C01', '203175700025_R05C01', '203175700025_
↪R06C01']
```

```
INFO:methylprep.files.manifests:Reading manifest file: MethylationEPIC_v-1-0_B4.
↪CoreColumns.csv
INFO:methylcheck.reports.qc_report:Beta MDS Plot
INFO:methylcheck.reports.qc_report:QC signal intensity plot
WARNING:methylcheck.qc_plot:Your poobah_values.pkl file contains missing values;␣
↪color coding will be inaccurate.
INFO:methylcheck.reports.qc_report:Control probes
INFO:methylcheck.reports.qc_report:Betas by probe type
INFO:methylprep.files.manifests:Reading manifest file: MethylationEPIC_v-1-0_B4.
↪CoreColumns.csv
```

[ ]:
```
# Once everything has been added, you need to close the PDF before you can read it on␣
↪disk.
my_report.pdf.close()
# look in the path folder for a PDF file called "some_file_name.pdf" -- it will be␣
↪similar to the charts
# returned by methylprep.run_qc()
```

You can customize your ReportPDF and even pass in customized tables like this:

```
# generate report
report = methylcheck.qc_report.ReportPDF(
    path=working.name,
    poobah_max_percent=10,
    pval_cutoff=0.01,
    title='QC Report',
    author='FOXO Biosciences',
    subject="QC Report",
    keywords="methylation array",
    outpath=working.name,
    filename=report_filename,
    poobah=True,
    on_lambda=True,
    custom_tables=custom_tables,
    debug=True,
    order=['beta_density_plot', 'mds', 'auto_qc',
        'M_vs_U', 'qc_signal_intensity', 'controls',
        'probe_types'],
)
```

(continues on next page)

(continued from previous page)

```
report.run_qc()
report.pdf.close()
```

Notes:

- 'on_lambda': if you are running this within an AWS lambda function in the cloud, the default paths of your manifest and other files will change. When True, `on_lambda` will allow you to specify and override paths to your input, output, and manifest files.
- 'path': where to read files from.
- 'outpath': in this example, `working.name` is a python `tempdir` folder in a lambda virtual environment. Everything is processed there but saved by moving to an S3 bucket.
- 'poobah': whether the QC should run on samples that have failed probes removed (recommended)
- custom_tables: passing in additional tables to the report. See the function's help for more details (e.g. `help(methylcheck.qc_report.ReportPDF)`).
- 'order': Option to specify the order of charts in the REPORT.

4.4 Custom QC with pOOBAH Vales

This tutorial is meant for those who want to have more customization to their quality control of beta values. Methylprep provides some automatic QC by default, but in this tutorial, we will go over how to do this manually, and with customizable parameters.

[1]:
```
import methylcheck
import pandas as pd
import numpy as np
```

Filepath of the processed files (Download and processing performed with Methylprep package)

[2]:
```
fpath = 'data/GPL13534/'
```

4.4.1 Load the Beta Values in a dataframe

The columns are each probe in the methylation array and the rows are each sample in the dataset. Note that if you want the dataframe in this orientation, you will need to transpose it.

The reason behind why we are using the `format='beta_csv'` in `methylcheck.load` is because this loads the raw beta values without any processing. By default, methylprep does some QC on the beta values automatically and saves those new beta values in `beta_values.pkl`. Specifically, it removes failed probes using Sesame pOOBAH method where a specific probe is classified as failed when the p-value >= 0.05.

If you want to use the pOOBAH to mask beta values yourself, you must specify `no_poobah=True`. Otherwise, it will mask them automatically when the CSV is loaded into a dataframe.

[22]:
```
betas = methylcheck.load('data/GPL13534', format='beta_csv', no_poobah=True).T
#betas.index.name = 'Samples'
print(betas.shape)
betas.head()
```

```
Files: 100%|| 121/121 [00:52<00:00,  2.30it/s]
INFO:methylcheck.load_processed:merging...
100%|| 121/121 [00:00<00:00, 692.51it/s]
```

```
(121, 485577)
```

[22]:
```
IlmnID             cg00000029  cg00000108  cg00000109  cg00000165  cg00000236  \
9996247040_R03C02       0.796       0.961       0.853       0.246       0.902
9996247040_R03C01       0.887       0.960       0.801       0.271       0.902
3998909005_R06C01       0.847       0.972       0.914       0.187       0.950
3998909005_R06C02       0.900       0.966       0.909       0.232       0.922
3998909206_R01C02       0.885       0.957       0.911       0.152       0.922

IlmnID             cg00000289  cg00000292  cg00000321  cg00000363  cg00000622  \
9996247040_R03C02       0.583       0.930       0.465       0.398       0.008
9996247040_R03C01       0.672       0.953       0.341       0.552       0.015
3998909005_R06C01       0.820       0.900       0.345       0.375       0.014
3998909005_R06C02       0.749       0.943       0.326       0.397       0.014
3998909206_R01C02       0.797       0.926       0.391       0.404       0.016

IlmnID             ...  rs7746156  rs798149  rs845016  rs877309  rs9292570  \
9996247040_R03C02  ...      0.468     0.374     0.079     0.016      0.978
9996247040_R03C01  ...      0.971     0.396     0.059     0.444      0.967
3998909005_R06C01  ...      0.512     0.018     0.469     0.539      0.019
3998909005_R06C02  ...      0.508     0.984     0.922     0.024      0.019
3998909206_R01C02  ...      0.045     0.977     0.491     0.531      0.480

IlmnID             rs9363764  rs939290  rs951295  rs966367  rs9839873
9996247040_R03C02      0.544     0.961     0.981     0.883      0.614
9996247040_R03C01      0.042     0.537     0.967     0.582      0.333
3998909005_R06C01      0.951     0.551     0.969     0.959      0.950
3998909005_R06C02      0.962     0.582     0.536     0.945      0.944
3998909206_R01C02      0.075     0.969     0.535     0.946      0.162

[5 rows x 485577 columns]
```

When loading the betas from the CSV, there are still control probes in your resulting dataframe. The cell below shows how to remove all of the control probes from you betas dataframe.

[23]:
```
rs_probes = betas.columns[betas.columns.str.startswith('rs')]
betas_nocontrol = betas.drop(rs_probes, axis=1)
print(betas_nocontrol.shape)
betas_nocontrol = betas_nocontrol.T[betas_nocontrol.index.sort_values()].T
betas_nocontrol
```

```
(121, 485512)
```

[23]:
```
IlmnID               cg00000029  cg00000108  cg00000109  cg00000165  \
100946230055_R04C01       0.864       0.971       0.925       0.288
100946230056_R04C01       0.854       0.978       0.930       0.215
100946230056_R04C02       0.879       0.958       0.866       0.257
101032570143_R04C02       0.837       0.968       0.911       0.334
101032570152_R04C01       0.813       0.971       0.928       0.164
...                         ...         ...         ...         ...
9996247054_R03C01         0.840       0.957       0.871       0.253
9996247054_R03C02         0.864       0.963       0.864       0.194
9996247055_R03C01         0.817       0.956       0.842       0.292
9996247055_R03C02         0.801       0.965       0.869       0.357
9996247056_R05C02         0.837       0.976       0.945       0.244
```

(continues on next page)

(continued from previous page)

```
IlmnID               cg00000236  cg00000289  cg00000292  cg00000321  \
100946230055_R04C01       0.935       0.654       0.945       0.378
100946230056_R04C01       0.932       0.639       0.971       0.421
100946230056_R04C02       0.899       0.604       0.971       0.191
101032570143_R04C02       0.918       0.765       0.951       0.435
101032570152_R04C01       0.934       0.810       0.955       0.358
...                         ...         ...         ...         ...
9996247054_R03C01         0.917       0.673       0.932       0.374
9996247054_R03C02         0.885       0.682       0.929       0.393
9996247055_R03C01         0.887       0.671       0.903       0.423
9996247055_R03C02         0.893       0.647       0.962       0.340
9996247056_R05C02         0.957       0.759       0.927       0.402

IlmnID               cg00000363  cg00000622  ...  ch.X.93511680F  \
100946230055_R04C01       0.468       0.010  ...           0.046
100946230056_R04C01       0.397       0.012  ...           0.034
100946230056_R04C02       0.560       0.012  ...           0.038
101032570143_R04C02       0.456       0.011  ...           0.034
101032570152_R04C01       0.372       0.011  ...           0.044
...                         ...         ...  ...             ...
9996247054_R03C01         0.374       0.012  ...           0.050
9996247054_R03C02         0.418       0.019  ...           0.047
9996247055_R03C01         0.490       0.014  ...           0.037
9996247055_R03C02         0.467       0.013  ...           0.046
9996247056_R05C02         0.454       0.013  ...           0.049

IlmnID               ch.X.938089F  ch.X.94051109R  ch.X.94260649R  \
100946230055_R04C01         0.036           0.031           0.171
100946230056_R04C01         0.040           0.034           0.105
100946230056_R04C02         0.057           0.045           0.253
101032570143_R04C02         0.098           0.058           0.329
101032570152_R04C01         0.058           0.036           0.171
...                           ...             ...             ...
9996247054_R03C01           0.037           0.040           0.234
9996247054_R03C02           0.048           0.029           0.155
9996247055_R03C01           0.042           0.026           0.191
9996247055_R03C02           0.045           0.025           0.157
9996247056_R05C02           0.030           0.025           0.052

IlmnID               ch.X.967194F  ch.X.97129969R  ch.X.97133160R  \
100946230055_R04C01         0.150           0.108           0.076
100946230056_R04C01         0.149           0.094           0.059
100946230056_R04C02         0.232           0.186           0.090
101032570143_R04C02         0.392           0.447           0.206
101032570152_R04C01         0.350           0.222           0.137
...                           ...             ...             ...
9996247054_R03C01           0.355           0.177           0.052
9996247054_R03C02           0.283           0.187           0.065
9996247055_R03C01           0.395           0.248           0.057
9996247055_R03C02           0.384           0.143           0.059
9996247056_R05C02           0.123           0.046           0.045

IlmnID               ch.X.97651759F  ch.X.97737721F  ch.X.98007042R
100946230055_R04C01           0.022           0.063           0.078
100946230056_R04C01           0.019           0.058           0.057
100946230056_R04C02           0.022           0.073           0.128
```

(continues on next page)

(continued from previous page)

```
101032570143_R04C02           0.032           0.070           0.142
101032570152_R04C01           0.031           0.061           0.152
...                             ...             ...             ...
9996247054_R03C01             0.029           0.055           0.099
9996247054_R03C02             0.037           0.086           0.180
9996247055_R03C01             0.031           0.067           0.132
9996247055_R03C02             0.024           0.055           0.097
9996247056_R05C02             0.026           0.065           0.063

[121 rows x 485512 columns]
```

4.4.2 Load p-values in a dataframe

This is reading in the pOOBAH values to a dataframe, and should have the same dimensions as the betas dataframe. Each cell in this dataframe is a p-value for each probe for a specific sample. If a p-value is >=0.05, then it's more likely that that specific probe for that sample failed. A failed probe means that the true probes signal is not istiguishable from the background fluorescence.

[19]:
```
p = pd.read_pickle('data/GPL13534/poobah_values.pkl').T
#p.index.name = 'Samples'
print(p.shape)
assert p.shape == betas_nocontrol.shape
print(f'Number of p-values >= 0.05: {(p>=0.05).sum().sum()}')
p = p.T[p.index.sort_values()].T
p.head()
```

```
(121, 485512)
Number of p-values >= 0.05: 1546688
```

[19]:
```
IlmnID               cg00000029  cg00000108  cg00000109  cg00000165  \
100946230055_R04C01       0.003       0.000       0.002       0.038
100946230056_R04C01       0.004       0.000       0.002       0.026
100946230056_R04C02       0.018       0.001       0.028       0.096
101032570143_R04C02       0.004       0.001       0.004       0.054
101032570152_R04C01       0.002       0.000       0.002       0.023

IlmnID               cg00000236  cg00000289  cg00000292  cg00000321  \
100946230055_R04C01       0.002       0.057         0.0       0.002
100946230056_R04C01       0.001       0.082         0.0       0.001
100946230056_R04C02       0.007       0.130         0.0       0.005
101032570143_R04C02       0.002       0.045         0.0       0.001
101032570152_R04C01       0.001       0.018         0.0       0.002

IlmnID               cg00000363  cg00000622  ...  ch.X.93511680F  \
100946230055_R04C01       0.001         0.0  ...             NaN
100946230056_R04C01       0.001         0.0  ...             NaN
100946230056_R04C02       0.002         0.0  ...             NaN
101032570143_R04C02       0.001         0.0  ...             NaN
101032570152_R04C01       0.001         0.0  ...             NaN

IlmnID               ch.X.938089F  ch.X.94051109R  ch.X.94260649R  \
100946230055_R04C01         0.006           0.004           0.073
100946230056_R04C01         0.004           0.003           0.038
100946230056_R04C02         0.010           0.006           0.127
101032570143_R04C02         0.025           0.014           0.241
101032570152_R04C01         0.014           0.006           0.086
```

(continues on next page)

(continued from previous page)

```
IlmnID               ch.X.967194F  ch.X.97129969R  ch.X.97133160R  \
100946230055_R04C01           NaN           0.037             NaN
100946230056_R04C01           NaN           0.035             NaN
100946230056_R04C02           NaN           0.082             NaN
101032570143_R04C02           NaN           0.535             NaN
101032570152_R04C01           NaN           0.117             NaN

IlmnID               ch.X.97651759F  ch.X.97737721F  ch.X.98007042R
100946230055_R04C01           0.001             NaN             NaN
100946230056_R04C01           0.001             NaN             NaN
100946230056_R04C02           0.001             NaN             NaN
101032570143_R04C02           0.003             NaN             NaN
101032570152_R04C01           0.003             NaN             NaN

[5 rows x 485512 columns]
```

4.4.3 Mask Beta values where probe fails

When the p-value of a probe for a specific sample >=0.05, it is more likely that the probe has failed, which means that the beta value for that probe may not be accurate. Because of this, it is a good idea to mask these beta values with a NULL value.

[24]:
```
cutoff = 0.05
betas_filtered = betas_nocontrol.mask((p>=cutoff), np.nan)

print(betas_filtered.shape)
print(f'Masked {betas_filtered.isna().sum().sum() - betas_nocontrol.isna().sum().
↪sum()} beta values')
betas_filtered
```

```
(121, 485512)
Masked 1546688 beta values
```

[24]:
```
IlmnID               cg00000029  cg00000108  cg00000109  cg00000165  \
100946230055_R04C01       0.864       0.971       0.925       0.288
100946230056_R04C01       0.854       0.978       0.930       0.215
100946230056_R04C02       0.879       0.958       0.866         NaN
101032570143_R04C02       0.837       0.968       0.911         NaN
101032570152_R04C01       0.813       0.971       0.928       0.164
...                         ...         ...         ...         ...
9996247054_R03C01         0.840       0.957       0.871       0.253
9996247054_R03C02         0.864       0.963       0.864       0.194
9996247055_R03C01         0.817       0.956       0.842         NaN
9996247055_R03C02         0.801       0.965       0.869         NaN
9996247056_R05C02         0.837       0.976       0.945       0.244

IlmnID               cg00000236  cg00000289  cg00000292  cg00000321  \
100946230055_R04C01       0.935         NaN       0.945       0.378
100946230056_R04C01       0.932         NaN       0.971       0.421
100946230056_R04C02       0.899         NaN       0.971       0.191
101032570143_R04C02       0.918       0.765       0.951       0.435
101032570152_R04C01       0.934       0.810       0.955       0.358
...                         ...         ...         ...         ...
9996247054_R03C01         0.917       0.673       0.932       0.374
9996247054_R03C02         0.885         NaN       0.929       0.393
```

(continues on next page)

(continued from previous page)

```
9996247055_R03C01         0.887       0.671       0.903       0.423
9996247055_R03C02         0.893         NaN       0.962       0.340
9996247056_R05C02         0.957       0.759       0.927       0.402

IlmnID               cg00000363  cg00000622  ...  ch.X.93511680F  \
100946230055_R04C01       0.468       0.010  ...           0.046
100946230056_R04C01       0.397       0.012  ...           0.034
100946230056_R04C02       0.560       0.012  ...           0.038
101032570143_R04C02       0.456       0.011  ...           0.034
101032570152_R04C01       0.372       0.011  ...           0.044
...                         ...         ...  ...             ...
9996247054_R03C01         0.374       0.012  ...           0.050
9996247054_R03C02         0.418       0.019  ...           0.047
9996247055_R03C01         0.490       0.014  ...           0.037
9996247055_R03C02         0.467       0.013  ...           0.046
9996247056_R05C02         0.454       0.013  ...           0.049

IlmnID               ch.X.938089F  ch.X.94051109R  ch.X.94260649R  \
100946230055_R04C01         0.036           0.031             NaN
100946230056_R04C01         0.040           0.034           0.105
100946230056_R04C02         0.057           0.045             NaN
101032570143_R04C02         0.098           0.058             NaN
101032570152_R04C01         0.058           0.036             NaN
...                           ...             ...             ...
9996247054_R03C01           0.037           0.040             NaN
9996247054_R03C02           0.048           0.029           0.155
9996247055_R03C01           0.042           0.026             NaN
9996247055_R03C02           0.045           0.025           0.157
9996247056_R05C02           0.030           0.025           0.052

IlmnID               ch.X.967194F  ch.X.97129969R  ch.X.97133160R  \
100946230055_R04C01         0.150           0.108           0.076
100946230056_R04C01         0.149           0.094           0.059
100946230056_R04C02         0.232             NaN           0.090
101032570143_R04C02         0.392             NaN           0.206
101032570152_R04C01         0.350             NaN           0.137
...                           ...             ...             ...
9996247054_R03C01           0.355             NaN           0.052
9996247054_R03C02           0.283             NaN           0.065
9996247055_R03C01           0.395             NaN           0.057
9996247055_R03C02           0.384           0.143           0.059
9996247056_R05C02           0.123           0.046           0.045

IlmnID               ch.X.97651759F  ch.X.97737721F  ch.X.98007042R
100946230055_R04C01           0.022           0.063           0.078
100946230056_R04C01           0.019           0.058           0.057
100946230056_R04C02           0.022           0.073           0.128
101032570143_R04C02           0.032           0.070           0.142
101032570152_R04C01           0.031           0.061           0.152
...                             ...             ...             ...
9996247054_R03C01             0.029           0.055           0.099
9996247054_R03C02             0.037           0.086           0.180
9996247055_R03C01             0.031           0.067           0.132
9996247055_R03C02             0.024           0.055           0.097
9996247056_R05C02             0.026           0.065           0.063

[121 rows x 485512 columns]
```

Start here if you have already masked your beta values based on p-values or had that done automatically

4.4.4 Remove Samples based on Percent or Number of Failed Probes

[40]:
```
percent_cutoff = 0.2 #use a percent in decimal format (20% = 0.2)
qc_betas = betas_filtered[~((betas_filtered.T.isna().sum() / betas_filtered.shape[1])
↪> 0.2)]

#if you want to remove samples based off a number threshold rather than a percentage,
↪use the following 2 lines:
#number_cutoff = 20000
#qc_betas = betas_filtered[~(betas_filtered.T.isna().sum() >=  number_cutoff)]

print(f'{betas_filtered.shape[0] - qc_betas.shape[0]} sample(s) removed because of
↪pOOBAH failure')
print(f'Sample(s) removed: {set(betas_filtered.index) - set(qc_betas.index)}')
print(qc_betas.shape)
qc_betas
```

```
1 sample(s) removed because of poobah failure
Sample(s) removed: {'101032570169_R04C02'}
(120, 485512)
```

[40]:
```
IlmnID               cg00000029  cg00000108  cg00000109  cg00000165  \
100946230055_R04C01       0.864       0.971       0.925       0.288
100946230056_R04C01       0.854       0.978       0.930       0.215
100946230056_R04C02       0.879       0.958       0.866         NaN
101032570143_R04C02       0.837       0.968       0.911         NaN
101032570152_R04C01       0.813       0.971       0.928       0.164
...                         ...         ...         ...         ...
9996247054_R03C01         0.840       0.957       0.871       0.253
9996247054_R03C02         0.864       0.963       0.864       0.194
9996247055_R03C01         0.817       0.956       0.842         NaN
9996247055_R03C02         0.801       0.965       0.869         NaN
9996247056_R05C02         0.837       0.976       0.945       0.244

IlmnID               cg00000236  cg00000289  cg00000292  cg00000321  \
100946230055_R04C01       0.935         NaN       0.945       0.378
100946230056_R04C01       0.932         NaN       0.971       0.421
100946230056_R04C02       0.899         NaN       0.971       0.191
101032570143_R04C02       0.918       0.765       0.951       0.435
101032570152_R04C01       0.934       0.810       0.955       0.358
...                         ...         ...         ...         ...
9996247054_R03C01         0.917       0.673       0.932       0.374
9996247054_R03C02         0.885         NaN       0.929       0.393
9996247055_R03C01         0.887       0.671       0.903       0.423
9996247055_R03C02         0.893         NaN       0.962       0.340
9996247056_R05C02         0.957       0.759       0.927       0.402

IlmnID               cg00000363  cg00000622  ...  ch.X.93511680F  \
100946230055_R04C01       0.468       0.010  ...           0.046
100946230056_R04C01       0.397       0.012  ...           0.034
100946230056_R04C02       0.560       0.012  ...           0.038
101032570143_R04C02       0.456       0.011  ...           0.034
101032570152_R04C01       0.372       0.011  ...           0.044
...                         ...         ...  ...             ...
9996247054_R03C01         0.374       0.012  ...           0.050
```

(continues on next page)

(continued from previous page)

```
9996247054_R03C02          0.418       0.019  ...           0.047
9996247055_R03C01          0.490       0.014  ...           0.037
9996247055_R03C02          0.467       0.013  ...           0.046
9996247056_R05C02          0.454       0.013  ...           0.049

IlmnID                ch.X.938089F  ch.X.94051109R  ch.X.94260649R  \
100946230055_R04C01          0.036           0.031             NaN
100946230056_R04C01          0.040           0.034           0.105
100946230056_R04C02          0.057           0.045             NaN
101032570143_R04C02          0.098           0.058             NaN
101032570152_R04C01          0.058           0.036             NaN
...                            ...             ...             ...
9996247054_R03C01            0.037           0.040             NaN
9996247054_R03C02            0.048           0.029           0.155
9996247055_R03C01            0.042           0.026             NaN
9996247055_R03C02            0.045           0.025           0.157
9996247056_R05C02            0.030           0.025           0.052

IlmnID                ch.X.967194F  ch.X.97129969R  ch.X.97133160R  \
100946230055_R04C01          0.150           0.108           0.076
100946230056_R04C01          0.149           0.094           0.059
100946230056_R04C02          0.232             NaN           0.090
101032570143_R04C02          0.392             NaN           0.206
101032570152_R04C01          0.350             NaN           0.137
...                            ...             ...             ...
9996247054_R03C01            0.355             NaN           0.052
9996247054_R03C02            0.283             NaN           0.065
9996247055_R03C01            0.395             NaN           0.057
9996247055_R03C02            0.384           0.143           0.059
9996247056_R05C02            0.123           0.046           0.045

IlmnID                ch.X.97651759F  ch.X.97737721F  ch.X.98007042R
100946230055_R04C01            0.022           0.063           0.078
100946230056_R04C01            0.019           0.058           0.057
100946230056_R04C02            0.022           0.073           0.128
101032570143_R04C02            0.032           0.070           0.142
101032570152_R04C01            0.031           0.061           0.152
...                              ...             ...             ...
9996247054_R03C01              0.029           0.055           0.099
9996247054_R03C02              0.037           0.086           0.180
9996247055_R03C01              0.031           0.067           0.132
9996247055_R03C02              0.024           0.055           0.097
9996247056_R05C02              0.026           0.065           0.063

[120 rows x 485512 columns]
```

4.4.5 Drop out Probes with a Percentage of NaNs

If you want to drop the probes with either all NaNs or a percentage of NaNs, use this code below. However, there are some scenarios where you will have to add back those probe columns, so only use this step if you have to.

[46]:
```
threshold = 0.95
final_betas = qc_betas.dropna(axis=1, thresh = int(threshold*qc_betas.shape[0]))
print(f'{qc_betas.shape[1] - final_betas.shape[1]} probe(s) removed because of NaNs')
#print(f'Sample(s) removed: {set(qc_betas.columns) - set(final_betas.columns)}')
↪#could be a long output
```

(continues on next page)

(continued from previous page)

```
final_betas
```

```
40600 probe(s) removed because of NaNs
```

[46]:
```
IlmnID               cg00000029  cg00000108  cg00000236  cg00000292  \
100946230055_R04C01       0.864       0.971       0.935       0.945
100946230056_R04C01       0.854       0.978       0.932       0.971
100946230056_R04C02       0.879       0.958       0.899       0.971
101032570143_R04C02       0.837       0.968       0.918       0.951
101032570152_R04C01       0.813       0.971       0.934       0.955
...                         ...         ...         ...         ...
9996247054_R03C01         0.840       0.957       0.917       0.932
9996247054_R03C02         0.864       0.963       0.885       0.929
9996247055_R03C01         0.817       0.956       0.887       0.903
9996247055_R03C02         0.801       0.965       0.893       0.962
9996247056_R05C02         0.837       0.976       0.957       0.927

IlmnID               cg00000321  cg00000363  cg00000622  cg00000658  \
100946230055_R04C01       0.378       0.468       0.010       0.862
100946230056_R04C01       0.421       0.397       0.012       0.907
100946230056_R04C02       0.191       0.560       0.012       0.871
101032570143_R04C02       0.435       0.456       0.011       0.912
101032570152_R04C01       0.358       0.372       0.011       0.835
...                         ...         ...         ...         ...
9996247054_R03C01         0.374       0.374       0.012       0.904
9996247054_R03C02         0.393       0.418       0.019       0.884
9996247055_R03C01         0.423       0.490       0.014       0.871
9996247055_R03C02         0.340       0.467       0.013       0.892
9996247056_R05C02         0.402       0.454       0.013       0.912

IlmnID               cg00000714  cg00000721  ...  ch.X.92543860F  \
100946230055_R04C01       0.249       0.936  ...           0.026
100946230056_R04C01       0.278       0.954  ...           0.022
100946230056_R04C02       0.235       0.923  ...           0.029
101032570143_R04C02       0.343       0.938  ...           0.033
101032570152_R04C01       0.333       0.957  ...           0.027
...                         ...         ...  ...             ...
9996247054_R03C01         0.411       0.923  ...           0.030
9996247054_R03C02         0.335       0.927  ...           0.033
9996247055_R03C01         0.391       0.894  ...           0.026
9996247055_R03C02         0.309       0.932  ...           0.033
9996247056_R05C02         0.299       0.945  ...           0.029

IlmnID               ch.X.92554290F  ch.X.93511680F  ch.X.938089F  \
100946230055_R04C01           0.023           0.046         0.036
100946230056_R04C01           0.023           0.034         0.040
100946230056_R04C02           0.022           0.038         0.057
101032570143_R04C02           0.038           0.034         0.098
101032570152_R04C01           0.032           0.044         0.058
...                             ...             ...           ...
9996247054_R03C01             0.028           0.050         0.037
9996247054_R03C02             0.037           0.047         0.048
9996247055_R03C01             0.032           0.037         0.042
9996247055_R03C02             0.028           0.046         0.045
9996247056_R05C02             0.030           0.049         0.030

IlmnID               ch.X.94051109R  ch.X.967194F  ch.X.97133160R  \
100946230055_R04C01           0.031         0.150           0.076
```

(continues on next page)

(continued from previous page)

```
100946230056_R04C01           0.034         0.149           0.059
100946230056_R04C02           0.045         0.232           0.090
101032570143_R04C02           0.058         0.392           0.206
101032570152_R04C01           0.036         0.350           0.137
...                             ...           ...             ...
9996247054_R03C01             0.040         0.355           0.052
9996247054_R03C02             0.029         0.283           0.065
9996247055_R03C01             0.026         0.395           0.057
9996247055_R03C02             0.025         0.384           0.059
9996247056_R05C02             0.025         0.123           0.045

IlmnID               ch.X.97651759F  ch.X.97737721F  ch.X.98007042R
100946230055_R04C01           0.022           0.063           0.078
100946230056_R04C01           0.019           0.058           0.057
100946230056_R04C02           0.022           0.073           0.128
101032570143_R04C02           0.032           0.070           0.142
101032570152_R04C01           0.031           0.061           0.152
...                             ...             ...             ...
9996247054_R03C01             0.029           0.055           0.099
9996247054_R03C02             0.037           0.086           0.180
9996247055_R03C01             0.031           0.067           0.132
9996247055_R03C02             0.024           0.055           0.097
9996247056_R05C02             0.026           0.065           0.063

[120 rows x 444912 columns]
```

Another way to tell if your sample is bad is to predict the sex of your samples and compare the predicted sex to the actual sex, if that information is available. If the predicted sex does not match the actual sex, this is an indicator that the sample needs to be investigated further, and could potentially be removed.

If you are planning on using your beta values for a machine learning model, you may want to filter out the sex probes to get rid of any sex bias in your model.

4.5 Outlier detection using Multidimensional Scaling (MDS)

Multidimensional scaling (MDS) is one method for comparing the similarity of samples–similar to PCA. MDS based on euclidean distance will also have identical results to PCA, because minimizing linear distance between points is effectively the same as maximizing linear correlation (the latter being what PCA does).

See StatQuest's very helpful video on MDS and PCoA for more details.

`methylcheck` allows users the option of filtering their data based on the results of MDS. The standard cut off is 1.5 standard deviations from the mean of the data, however users have the ability to adjust the cut off value if they are not satisfied by the filtering. After examining the MDS plot, press 'enter' to accept the cut-off as it is, or enter a new value and rerun the plot.

We will walk through an example of how to use MDS with this dataset from GEO: GSE111629. This is a large dataset (n=571) of patients with Parkinson's Disease (n=335) and a group of controls (n=237). These blood samples were run on Illumina's 450k arrays.

We downloaded the data from GEO and processed it with `methylprep` using the following command:

```
>>> python -m methylprep process -d <filepath> --all
```

WARNING: This is a huge dataset and `methylprep` will take +8 hours to process it. It will also eat up a lot of storage on your machine. We chose this data because it demonstrates the utility of the MDS function. We recommend

users pick a different dataset to follow along this example!

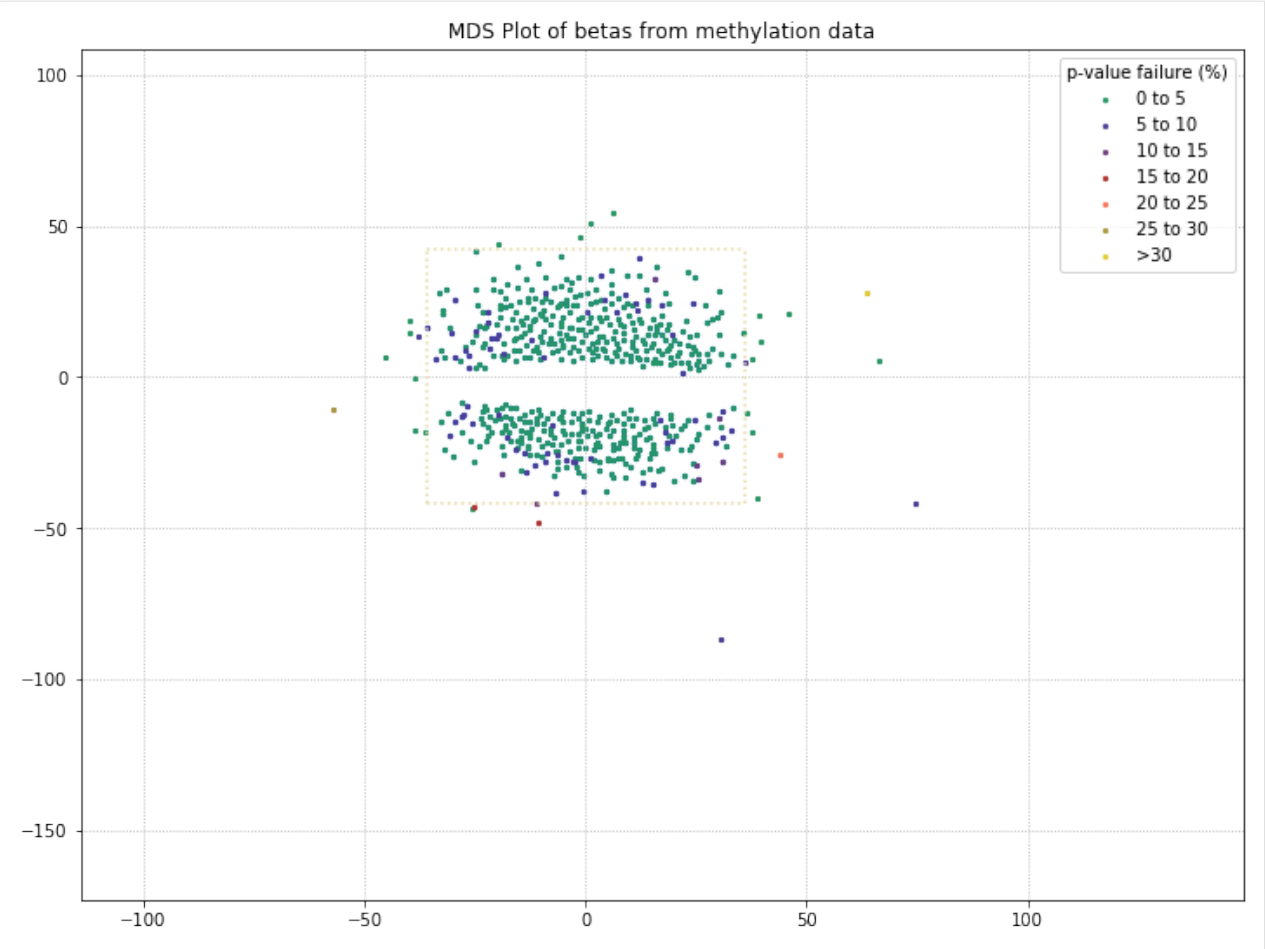
[1]:
```
import methylcheck
import pandas as pd
from pathlib import Path
filepath = Path('/Users/patriciagirardi/methylcheck_tutorial')
```

[2]:
```
df = pd.read_pickle('~/methylcheck_tutorial/beta_values.pkl')
metadata = pd.read_pickle('~/methylcheck_tutorial/GSE111629_GPL13534_meta_data.pkl')
metadata.head()
```

[2]:
```
        GSM_ID                          Sample_Name               source  \
0  GSM3035401  genomic DNA from 3999979001_R01C01  X3999979001_R01C01
1  GSM3035402  genomic DNA from 3999979001_R01C02  X3999979001_R01C02
2  GSM3035403  genomic DNA from 3999979001_R02C01  X3999979001_R02C01
3  GSM3035404  genomic DNA from 3999979001_R02C02  X3999979001_R02C02
4  GSM3035405  genomic DNA from 3999979001_R03C01  X3999979001_R03C01

   platform                                 title             disease state age  \
0  GPL13534  genomic DNA from 3999979001_R01C01  Parkinson's disease (PD)  74
1  GPL13534  genomic DNA from 3999979001_R01C02           PD-free control  73
2  GPL13534  genomic DNA from 3999979001_R02C01  Parkinson's disease (PD)  62
3  GPL13534  genomic DNA from 3999979001_R02C02           PD-free control  72
4  GPL13534  genomic DNA from 3999979001_R03C01           PD-free control  72

   gender  ethnicity       tissue          Sample_ID  Sentrix_ID  \
0  Female  Caucasian  whole blood  3999979001_R01C01  3999979001
1  Female  Caucasian  whole blood  3999979001_R01C02  3999979001
2    Male  Caucasian  whole blood  3999979001_R02C01  3999979001
3    Male  Caucasian  whole blood  3999979001_R02C02  3999979001
4  Female  Caucasian  whole blood  3999979001_R03C01  3999979001

  Sentrix_Position description
0           R01C01
1           R01C02
2           R02C01
3           R02C02
4           R03C01
```

[3]:
```
methylcheck.beta_mds_plot(df, filter_stdev=2, poobah=filepath)
```

```
INFO:methylcheck.samples.postprocessQC:81518.8 probe(s) [avg per sample] were missing␣
↪values and removed from MDS calculations; 485512 remaining.
INFO:numexpr.utils:NumExpr defaulting to 8 threads.
```




```
571 original samples; 542 after filtering
Your scale factor was: 2
```

[3]:

```
                3999979001_R01C01  3999979001_R01C02  3999979001_R02C01  \
IlmnID
cg00000029                  0.586              0.553              0.586
cg00000108                  0.978              0.974              0.972
cg00000109                  0.917              0.925              0.905
cg00000165                  0.221              0.208                NaN
cg00000236                  0.812              0.881              0.867
...                           ...                ...                ...
ch.X.97129969R                NaN              0.088                NaN
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.032              0.030              0.044
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

                3999979001_R02C02  3999979001_R03C01  3999979001_R03C02  \
IlmnID
cg00000029                  0.551              0.533              0.698
cg00000108                  0.973              0.968              0.973
cg00000109                  0.934              0.922              0.920
cg00000165                  0.185                NaN                NaN
cg00000236                  0.888              0.843              0.852
...                           ...                ...                ...
```

(continues on next page)

(continued from previous page)

```
ch.X.97129969R                NaN              0.112                NaN
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.031              0.029              0.043
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

               3999979001_R04C01  3999979001_R04C02  3999979001_R05C01  \
IlmnID
cg00000029                0.713              0.413              0.367
cg00000108                0.971              0.967              0.970
cg00000109                0.913              0.918              0.933
cg00000165                  NaN                NaN              0.160
cg00000236                0.889              0.815              0.809
...                         ...                ...                ...
ch.X.97129969R            0.102                NaN                NaN
ch.X.97133160R              NaN                NaN                NaN
ch.X.97651759F            0.024              0.038              0.038
ch.X.97737721F              NaN                NaN                NaN
ch.X.98007042R              NaN                NaN                NaN

               3999979001_R05C02  ...  9721367028_R01C02  9721367028_R02C02  \
IlmnID                            ...
cg00000029                0.655  ...              0.706              0.719
cg00000108                0.976  ...              0.969              0.974
cg00000109                0.923  ...              0.917              0.929
cg00000165                0.209  ...                NaN              0.209
cg00000236                0.887  ...              0.899              0.888
...                         ...  ...                ...                ...
ch.X.97129969R            0.118  ...                NaN              0.110
ch.X.97133160R              NaN  ...                NaN                NaN
ch.X.97651759F            0.024  ...              0.028              0.025
ch.X.97737721F              NaN  ...                NaN                NaN
ch.X.98007042R              NaN  ...                NaN                NaN

               9721367028_R03C01  9721367028_R03C02  9721367028_R04C01  \
IlmnID
cg00000029                0.521              0.643              0.531
cg00000108                0.970              0.973              0.972
cg00000109                0.930              0.924              0.930
cg00000165                  NaN                NaN                NaN
cg00000236                0.902              0.889              0.875
...                         ...                ...                ...
ch.X.97129969R            0.103                NaN                NaN
ch.X.97133160R              NaN                NaN                NaN
ch.X.97651759F            0.032              0.039              0.041
ch.X.97737721F              NaN                NaN                NaN
ch.X.98007042R              NaN                NaN                NaN

               9721367028_R04C02  9721367028_R05C01  9721367028_R05C02  \
IlmnID
cg00000029                0.578              0.518              0.606
cg00000108                0.968              0.969              0.975
cg00000109                0.905              0.893              0.912
cg00000165                  NaN                NaN                NaN
cg00000236                0.836              0.884              0.868
...                         ...                ...                ...
ch.X.97129969R              NaN              0.104              0.077
```

(continues on next page)

(continued from previous page)

```
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.042              0.027              0.027
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

                9721367028_R06C01  9721367028_R06C02
IlmnID
cg00000029                    NaN              0.532
cg00000108                  0.966              0.968
cg00000109                  0.908              0.912
cg00000165                    NaN                NaN
cg00000236                  0.882              0.874
...                           ...                ...
ch.X.97129969R                NaN                NaN
ch.X.97133160R                NaN                NaN
ch.X.97651759F              0.022              0.043
ch.X.97737721F                NaN                NaN
ch.X.98007042R                NaN                NaN

[485512 rows x 542 columns]
```

There are a few things to note with the MDS plot shown above. We've color-coded the samples in the plot by their poobah value failure rates; the samples with the highest failure rates (yellow, orange) tend to separate from the two clusters.

The other thing to note is that there are two clusters. With our color-coding scheme, we can only hope that these are the two groups (Parkinson's Disease patients vs. controls). So we need to adjust our color-coding to examine that possibility!

[4]:
```
metadata['disease state'].value_counts()
```

[4]:
```
Parkinson's disease (PD)    335
PD-free control             237
Name: disease state, dtype: int64
```

[5]:
```
disease_state = metadata[['Sample_ID', 'disease state']]
disease_labels = dict(zip(disease_state['Sample_ID'], disease_state['disease state']))
```

There's a small error in the metadata where one of the rows has an empty sample ID column, which will cause an error when we try to plot the mds based on the labels. See below how we call an empty string key from the labels dictionary and still get a value for the disease state. So we'll make sure to drop that key/value pair out from our dictionary before we continue.

[6]:
```
disease_labels['']
```

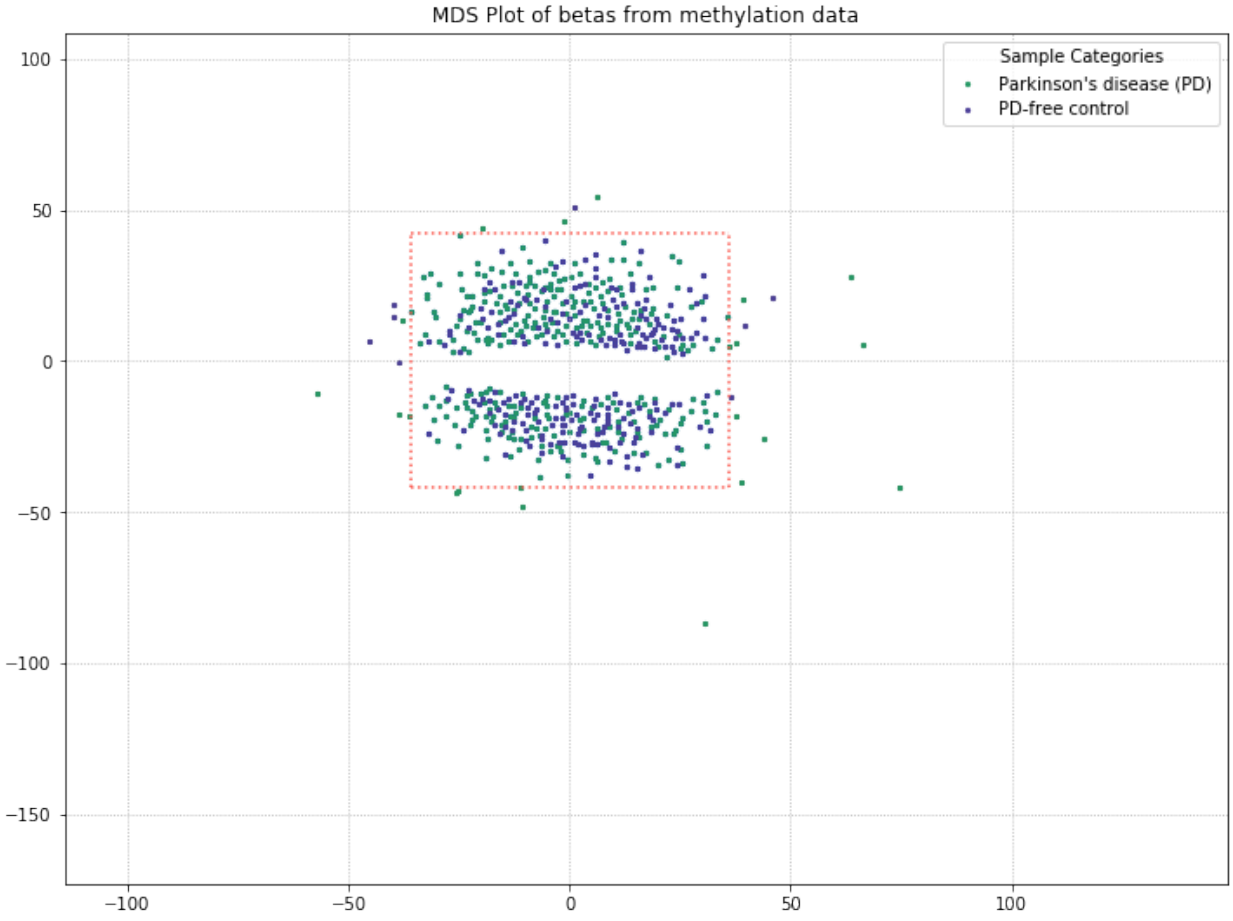
[6]:
```
"Parkinson's disease (PD)"
```

[7]:
```
del disease_labels['']
```

Now our dictionary won't error when we try to label samples in our MDS plot, so we can proceed with plotting the control vs. PD color-coding.

[8]:
```
methylcheck.beta_mds_plot(df, filter_stdev=2, labels=disease_labels)
```

```
INFO:methylcheck.samples.postprocessQC:81518.8 probe(s) [avg per sample] were missing␣
↪values and removed from MDS calculations; 485512 remaining.
```




```
571 original samples; 542 after filtering
Your scale factor was: 2
```

[8]:

IlmnID	3999979001_R01C01	3999979001_R01C02	3999979001_R02C01	\
cg00000029	0.586	0.553	0.586	
cg00000108	0.978	0.974	0.972	
cg00000109	0.917	0.925	0.905	
cg00000165	0.221	0.208	NaN	
cg00000236	0.812	0.881	0.867	
...	...	...	...	
ch.X.97129969R	NaN	0.088	NaN	
ch.X.97133160R	NaN	NaN	NaN	
ch.X.97651759F	0.032	0.030	0.044	
ch.X.97737721F	NaN	NaN	NaN	
ch.X.98007042R	NaN	NaN	NaN	

IlmnID	3999979001_R02C02	3999979001_R03C01	3999979001_R03C02	\
cg00000029	0.551	0.533	0.698	
cg00000108	0.973	0.968	0.973	
cg00000109	0.934	0.922	0.920	
cg00000165	0.185	NaN	NaN	
cg00000236	0.888	0.843	0.852	
...	...	...	...	

(continues on next page)

(continued from previous page)

```
ch.X.97129969R                NaN              0.112                NaN
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.031              0.029              0.043
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

               3999979001_R04C01  3999979001_R04C02  3999979001_R05C01  \
IlmnID
cg00000029                0.713              0.413              0.367
cg00000108                0.971              0.967              0.970
cg00000109                0.913              0.918              0.933
cg00000165                  NaN                NaN              0.160
cg00000236                0.889              0.815              0.809
...                         ...                ...                ...
ch.X.97129969R            0.102                NaN                NaN
ch.X.97133160R              NaN                NaN                NaN
ch.X.97651759F            0.024              0.038              0.038
ch.X.97737721F              NaN                NaN                NaN
ch.X.98007042R              NaN                NaN                NaN

               3999979001_R05C02  ...  9721367028_R01C02  9721367028_R02C02  \
IlmnID                            ...
cg00000029                0.655  ...              0.706              0.719
cg00000108                0.976  ...              0.969              0.974
cg00000109                0.923  ...              0.917              0.929
cg00000165                0.209  ...                NaN              0.209
cg00000236                0.887  ...              0.899              0.888
...                         ...  ...                ...                ...
ch.X.97129969R            0.118  ...                NaN              0.110
ch.X.97133160R              NaN  ...                NaN                NaN
ch.X.97651759F            0.024  ...              0.028              0.025
ch.X.97737721F              NaN  ...                NaN                NaN
ch.X.98007042R              NaN  ...                NaN                NaN

               9721367028_R03C01  9721367028_R03C02  9721367028_R04C01  \
IlmnID
cg00000029                0.521              0.643              0.531
cg00000108                0.970              0.973              0.972
cg00000109                0.930              0.924              0.930
cg00000165                  NaN                NaN                NaN
cg00000236                0.902              0.889              0.875
...                         ...                ...                ...
ch.X.97129969R            0.103                NaN                NaN
ch.X.97133160R              NaN                NaN                NaN
ch.X.97651759F            0.032              0.039              0.041
ch.X.97737721F              NaN                NaN                NaN
ch.X.98007042R              NaN                NaN                NaN

               9721367028_R04C02  9721367028_R05C01  9721367028_R05C02  \
IlmnID
cg00000029                0.578              0.518              0.606
cg00000108                0.968              0.969              0.975
cg00000109                0.905              0.893              0.912
cg00000165                  NaN                NaN                NaN
cg00000236                0.836              0.884              0.868
...                         ...                ...                ...
ch.X.97129969R              NaN              0.104              0.077
```

(continues on next page)

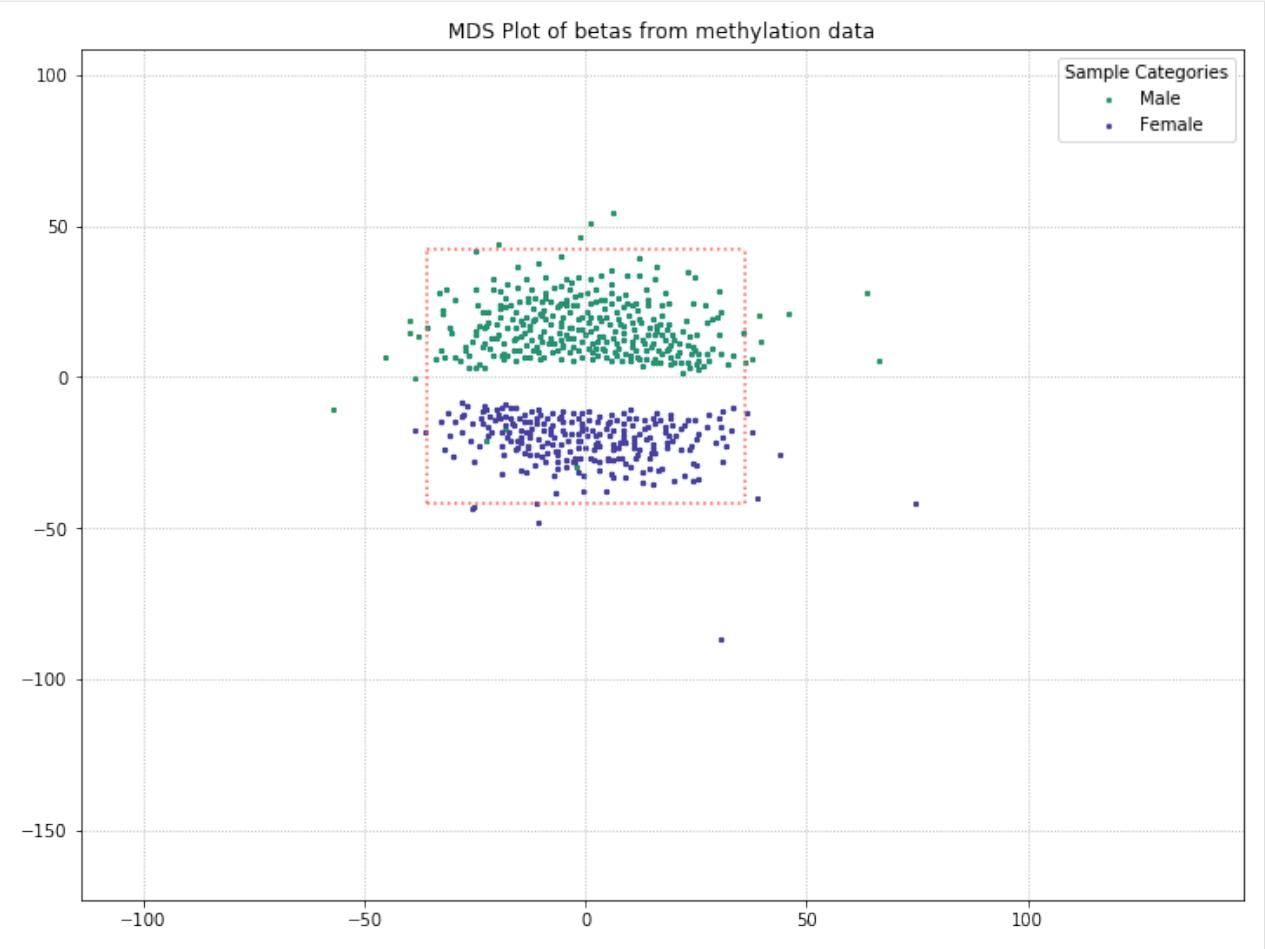
(continued from previous page)

```
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.042              0.027              0.027
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

                9721367028_R06C01  9721367028_R06C02
IlmnID
cg00000029                    NaN              0.532
cg00000108                  0.966              0.968
cg00000109                  0.908              0.912
cg00000165                    NaN                NaN
cg00000236                  0.882              0.874
...                           ...                ...
ch.X.97129969R                NaN                NaN
ch.X.97133160R                NaN                NaN
ch.X.97651759F              0.022              0.043
ch.X.97737721F                NaN                NaN
ch.X.98007042R                NaN                NaN

[485512 rows x 542 columns]
```

It appears that disease state isn't responsible for the clustering we're seeing here, based on these results. The PD and control patients are evenly interspersed among the two clusters. There might be some other attribute in the metadata responsible for this, though. The best candidate would be something with a binary result in this dataset, like patient gender. So let's examine what the MDS plot looks like when color-coding by patient gender.

[9]:
```
gender = metadata[['Sample_ID', 'gender']]
gender_labels = dict(zip(gender['Sample_ID'], gender['gender']))
del gender_labels['']

methylcheck.beta_mds_plot(df, filter_stdev=2, labels=gender_labels)
```

```
INFO:methylcheck.samples.postprocessQC:81518.8 probe(s) [avg per sample] were missing␣
↪values and removed from MDS calculations; 485512 remaining.
```




```
571 original samples; 542 after filtering
Your scale factor was: 2
```

[9]:

IlmnID	3999979001_R01C01	3999979001_R01C02	3999979001_R02C01	\
cg00000029	0.586	0.553	0.586	
cg00000108	0.978	0.974	0.972	
cg00000109	0.917	0.925	0.905	
cg00000165	0.221	0.208	NaN	
cg00000236	0.812	0.881	0.867	
...	...	...	...	
ch.X.97129969R	NaN	0.088	NaN	
ch.X.97133160R	NaN	NaN	NaN	
ch.X.97651759F	0.032	0.030	0.044	
ch.X.97737721F	NaN	NaN	NaN	
ch.X.98007042R	NaN	NaN	NaN	

IlmnID	3999979001_R02C02	3999979001_R03C01	3999979001_R03C02	\
cg00000029	0.551	0.533	0.698	
cg00000108	0.973	0.968	0.973	
cg00000109	0.934	0.922	0.920	
cg00000165	0.185	NaN	NaN	
cg00000236	0.888	0.843	0.852	
...	...	...	...	

(continues on next page)

(continued from previous page)

```
ch.X.97129969R                NaN              0.112                NaN
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.031              0.029              0.043
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

               3999979001_R04C01  3999979001_R04C02  3999979001_R05C01  \
IlmnID
cg00000029                 0.713              0.413              0.367
cg00000108                 0.971              0.967              0.970
cg00000109                 0.913              0.918              0.933
cg00000165                   NaN                NaN              0.160
cg00000236                 0.889              0.815              0.809
...                          ...                ...                ...
ch.X.97129969R             0.102                NaN                NaN
ch.X.97133160R               NaN                NaN                NaN
ch.X.97651759F             0.024              0.038              0.038
ch.X.97737721F               NaN                NaN                NaN
ch.X.98007042R               NaN                NaN                NaN

               3999979001_R05C02  ...  9721367028_R01C02  9721367028_R02C02  \
IlmnID                            ...
cg00000029                 0.655  ...              0.706              0.719
cg00000108                 0.976  ...              0.969              0.974
cg00000109                 0.923  ...              0.917              0.929
cg00000165                 0.209  ...                NaN              0.209
cg00000236                 0.887  ...              0.899              0.888
...                          ...  ...                ...                ...
ch.X.97129969R             0.118  ...                NaN              0.110
ch.X.97133160R               NaN  ...                NaN                NaN
ch.X.97651759F             0.024  ...              0.028              0.025
ch.X.97737721F               NaN  ...                NaN                NaN
ch.X.98007042R               NaN  ...                NaN                NaN

               9721367028_R03C01  9721367028_R03C02  9721367028_R04C01  \
IlmnID
cg00000029                 0.521              0.643              0.531
cg00000108                 0.970              0.973              0.972
cg00000109                 0.930              0.924              0.930
cg00000165                   NaN                NaN                NaN
cg00000236                 0.902              0.889              0.875
...                          ...                ...                ...
ch.X.97129969R             0.103                NaN                NaN
ch.X.97133160R               NaN                NaN                NaN
ch.X.97651759F             0.032              0.039              0.041
ch.X.97737721F               NaN                NaN                NaN
ch.X.98007042R               NaN                NaN                NaN

               9721367028_R04C02  9721367028_R05C01  9721367028_R05C02  \
IlmnID
cg00000029                 0.578              0.518              0.606
cg00000108                 0.968              0.969              0.975
cg00000109                 0.905              0.893              0.912
cg00000165                   NaN                NaN                NaN
cg00000236                 0.836              0.884              0.868
...                          ...                ...                ...
ch.X.97129969R               NaN              0.104              0.077
```

(continues on next page)

(continued from previous page)

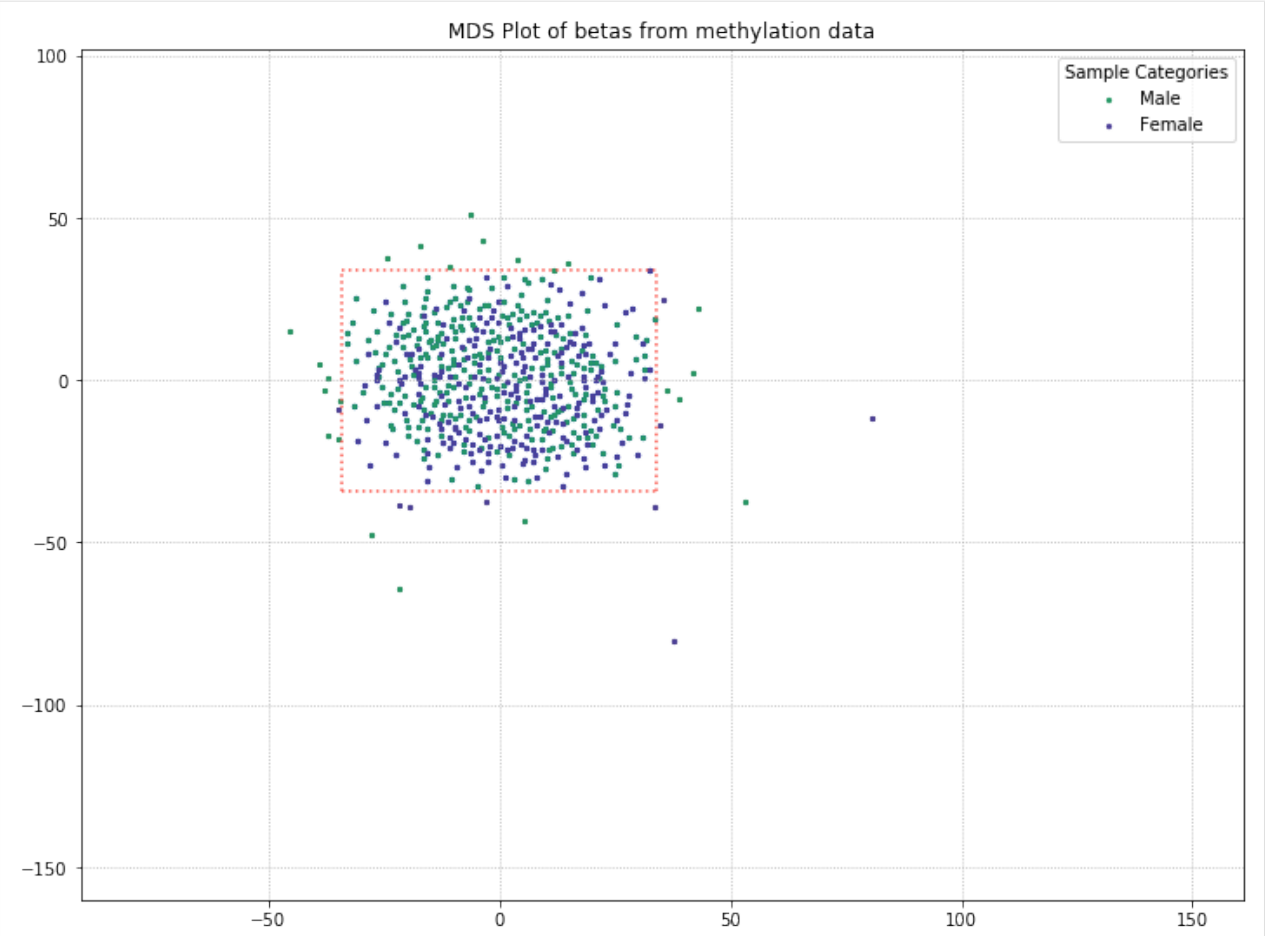
```
ch.X.97133160R                NaN                NaN                NaN
ch.X.97651759F              0.042              0.027              0.027
ch.X.97737721F                NaN                NaN                NaN
ch.X.98007042R                NaN                NaN                NaN

                9721367028_R06C01  9721367028_R06C02
IlmnID
cg00000029                    NaN              0.532
cg00000108                  0.966              0.968
cg00000109                  0.908              0.912
cg00000165                    NaN                NaN
cg00000236                  0.882              0.874
...                           ...                ...
ch.X.97129969R                NaN                NaN
ch.X.97133160R                NaN                NaN
ch.X.97651759F              0.022              0.043
ch.X.97737721F                NaN                NaN
ch.X.98007042R                NaN                NaN

[485512 rows x 542 columns]
```

It looks like this might be the culprit of the two clusters in the MDS plot! There are a few outliers, but for the most part, the two clusters are made up of one gender each.

Utilizing one of the functions covered in our Filtering Probes section, we can pull out the beta values from X/Y probes and just examine beta values from probes on the autosomal chromosomes. If MDS is picking up on differences between patient sex and clustering the patients accordingly, we might be able to get at an underlying difference by removing sex from the equation.

[11]:
```
no_sex_probes = methylcheck.exclude_sex_control_probes(df, array='450k', no_sex=True,␣
↪no_control=True)

methylcheck.beta_mds_plot(no_sex_probes, filter_stdev=2, labels=gender_labels)
```

```
INFO:methylcheck.samples.postprocessQC:79013.0 probe(s) [avg per sample] were missing␣
↪values and removed from MDS calculations; 473864 remaining.
```




```
571 original samples; 539 after filtering
Your scale factor was: 2
```

[11]:

```
                3999979001_R01C01  3999979001_R01C02  3999979001_R02C01  \
IlmnID
cg00000029                  0.586              0.553              0.586
cg00000108                  0.978              0.974              0.972
cg00000109                  0.917              0.925              0.905
cg00000165                  0.221              0.208                NaN
cg00000236                  0.812              0.881              0.867
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                3999979001_R02C02  3999979001_R03C01  3999979001_R03C02  \
IlmnID
cg00000029                  0.551              0.533              0.698
cg00000108                  0.973              0.968              0.973
cg00000109                  0.934              0.922              0.920
cg00000165                  0.185                NaN                NaN
cg00000236                  0.888              0.843              0.852
...                           ...                ...                ...
```

(continues on next page)

(continued from previous page)

```
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                3999979001_R04C01  3999979001_R05C01  3999979001_R05C02  \
IlmnID
cg00000029                  0.713              0.367              0.655
cg00000108                  0.971              0.970              0.976
cg00000109                  0.913              0.933              0.923
cg00000165                    NaN              0.160              0.209
cg00000236                  0.889              0.809              0.887
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                3999979001_R06C01  ...  9721367028_R01C02  9721367028_R02C02  \
IlmnID                             ...
cg00000029                  0.695  ...              0.706              0.719
cg00000108                  0.976  ...              0.969              0.974
cg00000109                  0.925  ...              0.917              0.929
cg00000165                  0.265  ...                NaN              0.209
cg00000236                  0.880  ...              0.899              0.888
...                           ...  ...                ...                ...
ch.9.98937537R                NaN  ...                NaN                NaN
ch.9.98957343R                NaN  ...                NaN                NaN
ch.9.98959675F                NaN  ...                NaN                NaN
ch.9.98989607R                NaN  ...                NaN                NaN
ch.9.991104F                  NaN  ...                NaN                NaN

                9721367028_R03C01  9721367028_R03C02  9721367028_R04C01  \
IlmnID
cg00000029                  0.521              0.643              0.531
cg00000108                  0.970              0.973              0.972
cg00000109                  0.930              0.924              0.930
cg00000165                    NaN                NaN                NaN
cg00000236                  0.902              0.889              0.875
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                9721367028_R04C02  9721367028_R05C01  9721367028_R05C02  \
IlmnID
cg00000029                  0.578              0.518              0.606
cg00000108                  0.968              0.969              0.975
cg00000109                  0.905              0.893              0.912
cg00000165                    NaN                NaN                NaN
cg00000236                  0.836              0.884              0.868
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
```

(continues on next page)

(continued from previous page)

```
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

               9721367028_R06C01  9721367028_R06C02
IlmnID
cg00000029                   NaN              0.532
cg00000108                 0.966              0.968
cg00000109                 0.908              0.912
cg00000165                   NaN                NaN
cg00000236                 0.882              0.874
...                          ...                ...
ch.9.98937537R               NaN                NaN
ch.9.98957343R               NaN                NaN
ch.9.98959675F               NaN                NaN
ch.9.98989607R               NaN                NaN
ch.9.991104F                 NaN                NaN

[473864 rows x 539 columns]
```

We lost the separation between clusters as a result of pulling out sex probes. There's still a chance that there is some kind of grouping going on with PD/control patients (for example, the top half of the data is all PD patients, while the bottom half is all control patients). So we'll color-code by disease state to verify whether or not there's any grouping going on.

[12]:
```
methylcheck.beta_mds_plot(no_sex_probes, filter_stdev=2, labels=disease_labels)
```

```
INFO:methylcheck.samples.postprocessQC:79013.0 probe(s) [avg per sample] were missing␣
↪values and removed from MDS calculations; 473864 remaining.
```


```
571 original samples; 539 after filtering
Your scale factor was: 2
```

[12]:

IlmnID	3999979001_R01C01	3999979001_R01C02	3999979001_R02C01	\
cg00000029	0.586	0.553	0.586	
cg00000108	0.978	0.974	0.972	
cg00000109	0.917	0.925	0.905	
cg00000165	0.221	0.208	NaN	
cg00000236	0.812	0.881	0.867	
...	...	...	...	
ch.9.98937537R	NaN	NaN	NaN	
ch.9.98957343R	NaN	NaN	NaN	
ch.9.98959675F	NaN	NaN	NaN	
ch.9.98989607R	NaN	NaN	NaN	
ch.9.991104F	NaN	NaN	NaN	

IlmnID	3999979001_R02C02	3999979001_R03C01	3999979001_R03C02	\
cg00000029	0.551	0.533	0.698	
cg00000108	0.973	0.968	0.973	
cg00000109	0.934	0.922	0.920	
cg00000165	0.185	NaN	NaN	
cg00000236	0.888	0.843	0.852	
...	...	...	...	

(continues on next page)

(continued from previous page)

```
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                3999979001_R04C01  3999979001_R05C01  3999979001_R05C02  \
IlmnID
cg00000029                  0.713              0.367              0.655
cg00000108                  0.971              0.970              0.976
cg00000109                  0.913              0.933              0.923
cg00000165                    NaN              0.160              0.209
cg00000236                  0.889              0.809              0.887
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                3999979001_R06C01  ...  9721367028_R01C02  9721367028_R02C02  \
IlmnID                             ...
cg00000029                  0.695  ...              0.706              0.719
cg00000108                  0.976  ...              0.969              0.974
cg00000109                  0.925  ...              0.917              0.929
cg00000165                  0.265  ...                NaN              0.209
cg00000236                  0.880  ...              0.899              0.888
...                           ...  ...                ...                ...
ch.9.98937537R                NaN  ...                NaN                NaN
ch.9.98957343R                NaN  ...                NaN                NaN
ch.9.98959675F                NaN  ...                NaN                NaN
ch.9.98989607R                NaN  ...                NaN                NaN
ch.9.991104F                  NaN  ...                NaN                NaN

                9721367028_R03C01  9721367028_R03C02  9721367028_R04C01  \
IlmnID
cg00000029                  0.521              0.643              0.531
cg00000108                  0.970              0.973              0.972
cg00000109                  0.930              0.924              0.930
cg00000165                    NaN                NaN                NaN
cg00000236                  0.902              0.889              0.875
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                9721367028_R04C02  9721367028_R05C01  9721367028_R05C02  \
IlmnID
cg00000029                  0.578              0.518              0.606
cg00000108                  0.968              0.969              0.975
cg00000109                  0.905              0.893              0.912
cg00000165                    NaN                NaN                NaN
cg00000236                  0.836              0.884              0.868
...                           ...                ...                ...
ch.9.98937537R                NaN                NaN                NaN
```

(continues on next page)

(continued from previous page)

```
ch.9.98957343R                NaN                NaN                NaN
ch.9.98959675F                NaN                NaN                NaN
ch.9.98989607R                NaN                NaN                NaN
ch.9.991104F                  NaN                NaN                NaN

                9721367028_R06C01  9721367028_R06C02
IlmnID
cg00000029                    NaN              0.532
cg00000108                  0.966              0.968
cg00000109                  0.908              0.912
cg00000165                    NaN                NaN
cg00000236                  0.882              0.874
...                           ...                ...
ch.9.98937537R                NaN                NaN
ch.9.98957343R                NaN                NaN
ch.9.98959675F                NaN                NaN
ch.9.98989607R                NaN                NaN
ch.9.991104F                  NaN                NaN

[473864 rows x 539 columns]
```

It appears that the differences between these two disease states aren't being picked up by MDS in a meaningful way. We'll do one more MDS plot on these filtered probes to check how many of those outlier samples have high detection p-value failure rates.

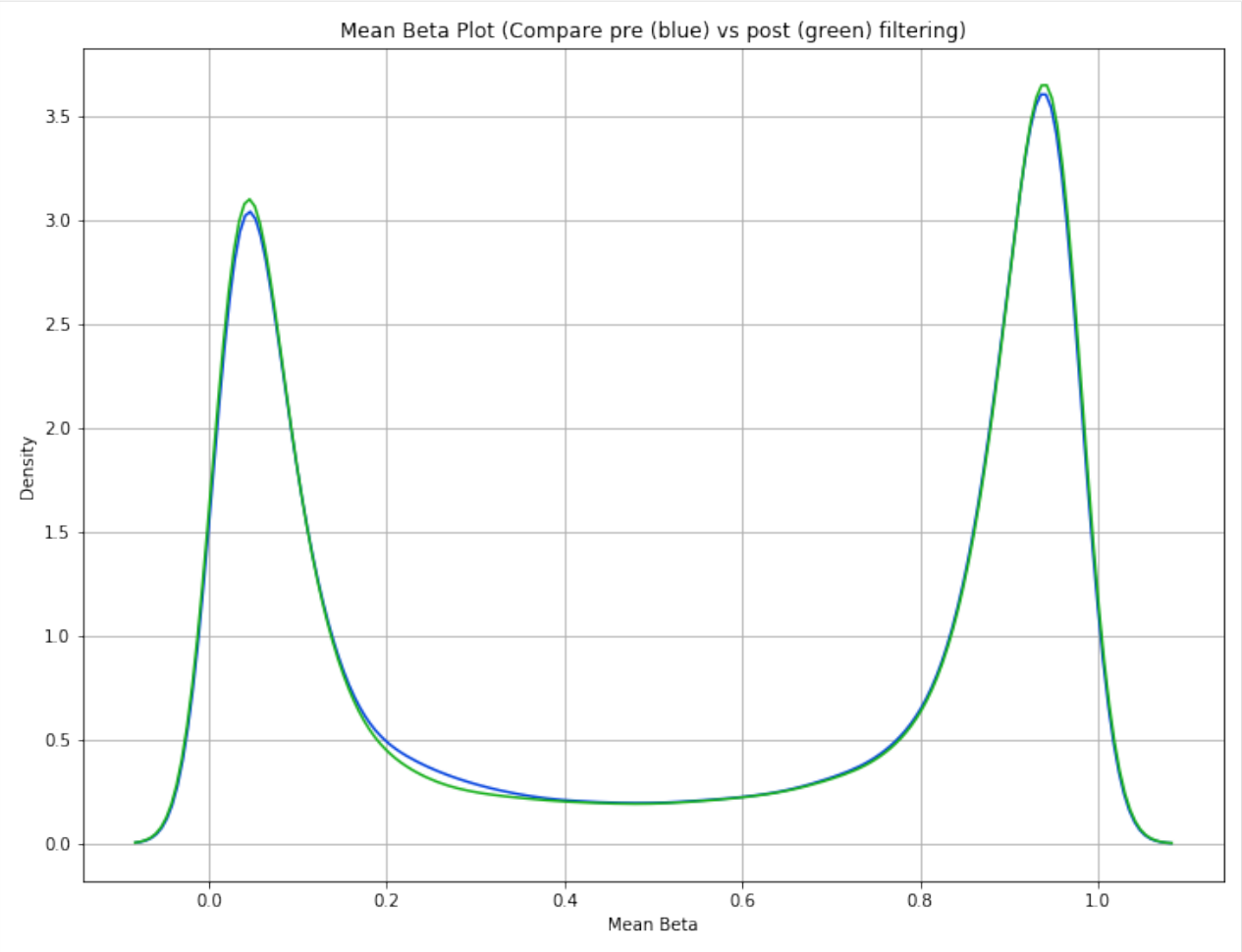
Also, we can define a variable here to access the samples that pass after they've been filtered by MDS. Then, we can compare that beta distribution to the original beta distribution to see how the MDS filtering affected it.

[13]:
```
mds_filtered = methylcheck.beta_mds_plot(no_sex_probes, filter_stdev=1.5,␣
↪poobah=filepath)
```

```
INFO:methylcheck.samples.postprocessQC:79013.0 probe(s) [avg per sample] were missing␣
↪values and removed from MDS calculations; 473864 remaining.
```


```
571 original samples; 448 after filtering
Your scale factor was: 1.5
```

[14]:
```
methylcheck.mean_beta_compare(df, mds_filtered)
```



The difference is a bit subtle since even the low quality data in this dataset is relatively high quality, but you can see that the green curve (MDS filtered samples) has higher and tighter peaks than the blue curve (unfiltered data). This means we were able to eliminate some of the samples that had a lot of beta values falling in that midrange between the peaks.

4.6 API Reference

`methylcheck.cli`
`methylcheck.run_pipeline`
`methylcheck.run_qc`
`methylcheck.read_geo`
`methylcheck.load`
`methylcheck.load_both`
`methylcheck.qc_signal_intensity`
`methylcheck.plot_M_vs_U`
`methylcheck.plot_controls`
`methylcheck.plot_beta_by_type`
`methylcheck.probes`
`methylcheck.list_problem_probes`

Continued on next page

Table 1 – continued from previous page

`methylcheck.exclude_probes`
`methylcheck.exclude_sex_control_probes`
`methylcheck.drop_nan_probes`
`methylcheck.samples`
`methylcheck.sample_plot`
`methylcheck.beta_density_plot`
`methylcheck.mean_beta_plot`
`methylcheck.mean_beta_compare`
`methylcheck.beta_mds_plot`
`methylcheck.combine_mds`
`methylcheck.cumulative_sum_beta_distribution`
`methylcheck.predict`
`methylcheck.get_sex`
`methylcheck.assign`

4.6.1 Loading Data

4.6.2 `ReportPDF` Report Builder class

4.6.3 Run QC pipeline

4.6.4 filtering probes

4.6.5 plotting functions

4.6.6 sex prediction

4.7 Release History

4.7.1 v0.8.5

- functions support .parquet files as input (produced by methylprep pipeline v1.7.0 or higher)
- methylcheck.load() includes a 'control' format that returns a dictionary of dataframes, keyed to sample names.
- consistent flexible function inputs: Any function that accepted a filepath or dataframe as first parameter now accepts either of these interchangeably, except for loading functions and Report classes; these cannot handle dataframe inputs: `ControlsReport`; `ReportPDF`; `load()`; `load_both()`; `combine_mds()`.
- improved .load speed

4.7.2 v0.8.4

- get_sex bug fixes; supports plots, returning figure, returning labels, or returning predicted sex dataframe
- add testing via github actions
- updated documentation

4.7.3 v0.8.2

- added support for sample sheets with the legacy Illumina [Header] ... [Data] format. This requires `methylprep` be installed for the controls report to run now.

4.7.4 v0.8.1

- .load gives clearer error when loading beta values from CSVs ('beta_csv') if probe names are not unique, and returns a list of series for each sample when indeces fail to merge (pandas.concat)
- .beta_mds_plot() can now suppress the interactive portion and still display plots, using `silent=True` and `plot=True` (`plot` is a new kwarg, and defaults to `True`). Previously `silent` mode would suppress both prompts and plot display. Change in behavior: `silent` mode will not disable plotting. Must also include `plot=False` for that.

4.7.5 v0.8.0

- Fixed bug in `.load` that requires `tqdm` >= 4.61.2
- Added more detailed error message on `.load`; it cannot load and merge two meth/unmeth dataframes with redundant probe names.

4.7.6 v0.7.9

- `ReportPDF` accepts 'poobah_colormap' kwarg to feed in beta_mds_plot colormap.
- `ReportPDF` custom tables: You can insert your custom table on the first page by specifying 'order_after' == None.
- `beta_mds_plot` `palette` can now be any matplotlib colormap name. Defaults to 'magma' if not specified. The palette is only used to color-code poobah failure rates, if the poobah file path is specified.
- `beta_mds_plot` new kwarg `extend_poobah_range`: Default (True) shows 7 colors for poobah failure rates. If False, will show only 5.

4.7.7 v0.7.6

- Reading IDATs loading bar didn't work correctly, showed up after loading.
- Fixed error/logging messages:
  - exclude_sex_control_probes() said 916 control probes were removed, then said "it appears your sample had no control probes"
  - Erroneous message about missing values in poobah file: "color coding may be inaccurate."
  - Filtering probes info message said there were N samples when it meant probes.
  - methylprep.download.build_composite_dataset() Process time was negative.
- Target Removal and Staining graphs in plot_controls() had unreadable X-axis sample names. Labels are suppressed when showing more than 30 samples.
- methylcheck.detect_array() sometimes returned array types in wrong case. All functions expect lowercase array types now.
  - resolves exclude_sex_control_probes bugs.

- run_qc() and get_sex() did not recognize poobah_values.pkl on MacOS when using "~" in the filepath.
- methylcheck.problem_probe_reasons() lists probes matching any/all criteria when passing in no arguments, as documented
- get_sex() understands samplesheet 'm' and 'f' when not capitalized now.
- Load_both: always returns dataframe with probes in rows now, like .load() does.
- plot_M_vs_U now loads the noob_meth_values.pkl files if noob=True and files are found; otherwise it uses whatever meth/unmeth data is available.
- Methylcheck.qc_plot.qc_signal_intensity returns a dictionary of data about good/bad samples based on signal intensity. Previously it was only returning this if 'plot' was False.
- controls_report() bug fixed: methylprep was producing samplesheet meta data pickles that contained Sample_ID twice, because the GEO series_matrix files had this data appear twice. This broke the report, but this case is caught and avoided now. controls_report() will recognize a wider array of samplesheet filenames now; anything with 'samplesheet' or 'meta_data' in the filename.

4.7.8 v0.7.5

- added 'methylcheck report' CLI option to create a ReportPDF
- updated documentation
- minor bug fixes in read_geo()
  - qc_plot() now handles mouse probe type differently
  - handles importing from multiple pandas versions correctly
  - read_geo can open series_matrix.txt files now

4.7.9 v0.7.4

- fixed big where csv data_files were not included in pypi

4.7.10 v0.7.3

- Improved ReportPDF custom tables option
  - if fields are too long, it will truncate them or auto scale the font size smaller to fit on page.

4.7.11 v0.7.2

- added GCT score to controls_report() used in the ReportPDF class.
- ReportPDF changes
  - uses noob_meth/unmeth instead of raw, uncorrected meth/unmeth values for GCT and U vs M plot
  - inverted poobah table to report percent passing (instead of failing) probes per sample
  - this changed input from 'poobah_max_percent' (default 5%) to 'poobah_min_percent', (default 80%)
  - M_vs_U not included by default, because redundant with qc_signal_intensity
  - M_vs_U compare=True now labels each sample and has legend, so you can see effect of NOOB+dye correction on batch

  - added poobah color-coding to MDS plot
- get_sex improved plotting
  - will read poobah data and size sample points according to percent of failed probes
  - save plots, or return fig, and more options now

4.7.12 v0.7.1

- Added a controls_report() function that creates a spreadsheet summary of control probe performance.
- **New unit test coverage. Note that because methylprep v1.4.0 changes processing, the results will change slightly** to match `sesame` instead of `minfi`, with nonlinear-dye-bias correction and infer-type-I-probe-switching.
- changed org name from FoxoBioScience to FoxoTech

4.7.13 v0.7.0

- Illumina Mouse Array Support
- Complete rewrite of documentation
- qc_signal_intensity and plot_M_vs_U have additional options, including superimposing poobah (percent probe failures per sample) on the plot coloring.
- .load will work on control_probes.pkl and mouse_probes.pkl files (with alt structure: dictionary of dataframe)
- .sample_plot uses "best" legend positioning now, because it was not fitting on screen with prev settings.

4.7.14 v0.6.4

- get_sex() function returns a dataframe that also includes percent of X and Y probes that failed p-value-probe detection, as an indication of whether the predicted sex is reliable.
- better unit test coverage of predictions, load, load_both, and container_to_pkl functions
- fixed bug in load('meth_df')

4.7.15 v0.6.3

- fixed bug in detect_array() where it labeled EPIC+ as EPIC

4.7.16 v0.6.2

- minor fixes to load() and read_geo()
- exclude_probes() accepts problem_probes criteria as alternate way to specify probes.
  - Exclude probes from a df of samples. Use list_problem_probes() to obtain a list of probes (or pass in the names of 'Criteria' from problem probes), then pass that in as a probe_list along with the dataframe of beta values (array)
- load_processed now has a –no_filter=False option that will remove probes that failed p-value detection, if passing in beta_values.pkl and poobah_values.pkl files.
- load() now handles gzipped files the same way (so .pkl.gz or .csv.gz OK as file or folder inputs)

- seaborn v0.10 –> v0.11 deprecrated the distplot() function that was used heavily. So now this employs kdeplot() in its place, with similar results.

4.7.17 v0.6.1

- exposed more beta_density_plot parameters, so it can be used to make a QC plot (highlighting one or several samples within a larger batch, and graying out the others in the plot).

4.7.18 v0.6.0

- improved read_geo() function, for downloading GEO methylation data sets and parsing meta_data from projects.
- changed org name from life-epigenetics to FoxoBioScience on Github.

4.7.19 v0.5.9

- qc_plot bug fixes -99

4.7.20 v0.5.7

- -99 bug in negative controls fixed

4.7.21 v0.5.4

- tweaking custom-tables in ReportPDF

4.7.22 v0.5.2

- ReportPDF.run_qc() supports on_lambda, and any functions that require .methylprep_manifest_files can be set to look for manifests in /tmp using on_lambda=True

4.7.23 v0.5.1

- sklearn now optional for MDS

4.7.24 v0.5.0

- adds kwargs to functions for silent processing returning figure objects, and a report_pdf class that can run QC and generate a PDF report.
- added **version**
- p-value probe detection
- hdbscan clustering functions
- more QC methods testing

4.7.25 v0.4.0

- more tests, smart about df orientation, and re-organized files
- added read_geo() for processed datafiles, and unit tests for it. Works with txt,csv,xlsx,pkl files
- read_geo() docs
- debugged filters.list_problem_probes:
- updated the docs to have correct spelling for refs/reasons.
- added a function that lets you see more detail on the probes and reasons/pubs criteria
- added more genome studio QC functions,
  - improved .load function (but not consolidated through methyl-suite yet)
  - function .assign() for manually categorizing samples
  - unit testing on the predict.sex function
  - get_sex() prediction
- consolidated data loading for functions and uses fastest option

CHAPTER 5

Indices and tables

- genindex