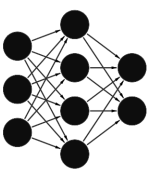
PREDICTION OF PATHOGENIC SNV

Prof. Giorgio Valentini
6 CFU

Luca Cappelletti

Course Project
Year 2017/2018



IT Master Degree
Universiy of Milan
Italy
29 giugno 2018

Indice

Parte I

Dataset

1 Data points

First we begin looking at the dataset, the distributions of the given metrics and the statistical analysis of these data points.

1.1 Retrieving the dataset

The dataset can be downloaded from https://homes.di.unimi.it/valentini/ProgettoBioinformatica1718/data/.

1.2 Composition

1.2.1 Training dataset

In the training dataset there are 981388 data points, each one comprised of 26 metrics. The first 356 are pathogenic and all the others are negative.

1.2.2 Testing dataset

In the test dataset there are 19018 data points, still each one comprised of 26 metrics. The first 40 are pathogenic and the following are negative.

2 Metrics

2.1 How the graphs are realized

All the graphs are in triples: positives, negatives and mixed.

The normalization is done, as usual, in the following way:

$$m' = \frac{\text{metric} - \mathbb{E}\,(\text{metric values})}{\max\{\text{metric values}\} - \min\{\text{metric values}\}}$$

Figura 2.1: Input normalization

2.1.1 Metric sample distribution

Are realized by calculating the frequencies and estimating the density distributions parameters via MLE.

2.1.2 Plot graphs

Plot graphs are realized by sorting the values of the single metrics.

2.1.3 Normalized plot graphs

Are realized by sorting the values of the metric, with the domain and codomain normalized.

2.2 CpGobsExp

2.2.1 Metric sample distribution

The data points seem to follow a **Beta** distribution.

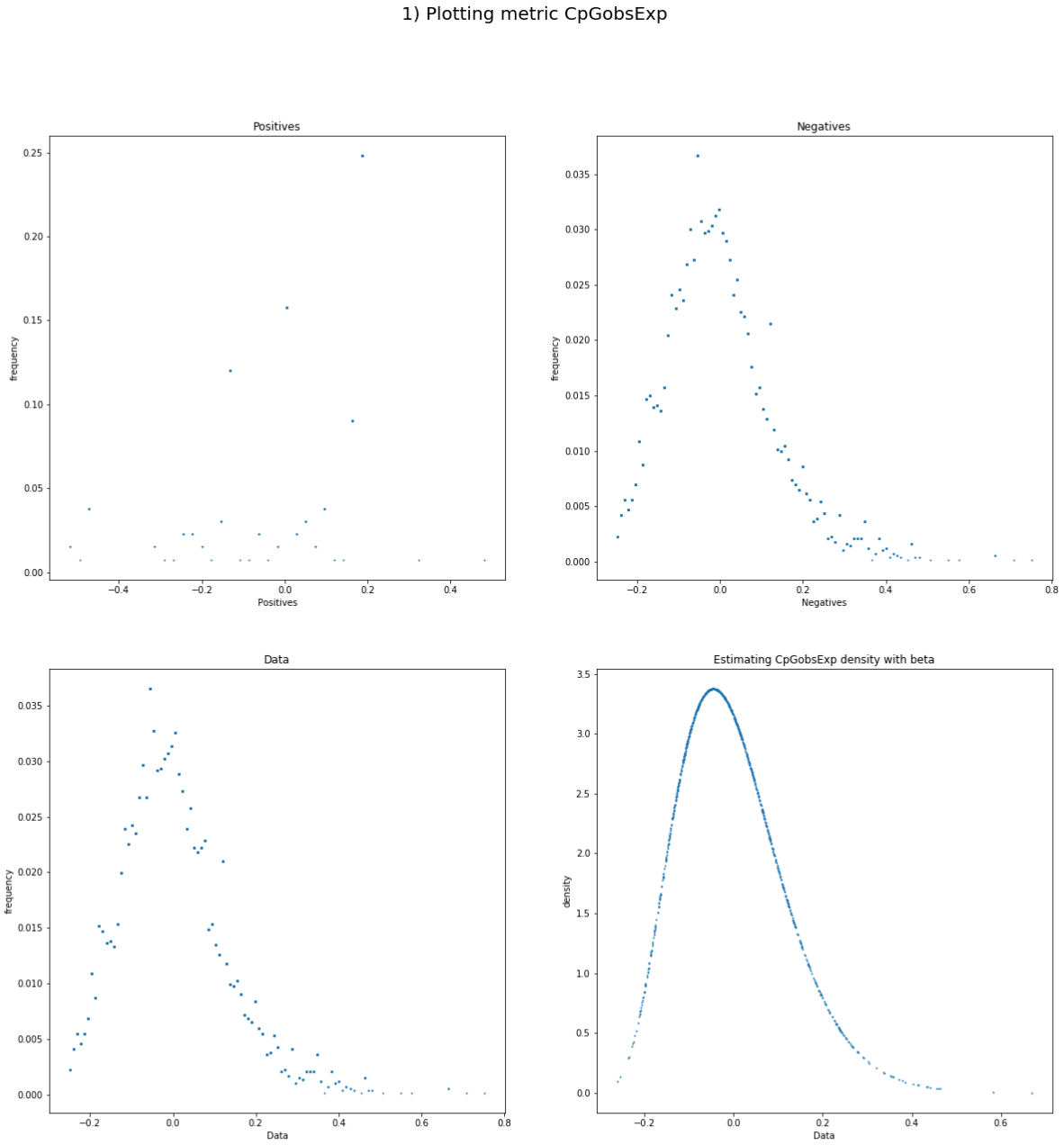


Figura 2.2: Sampling distribution of metric CpGobsExp

2.2.2 Metric values

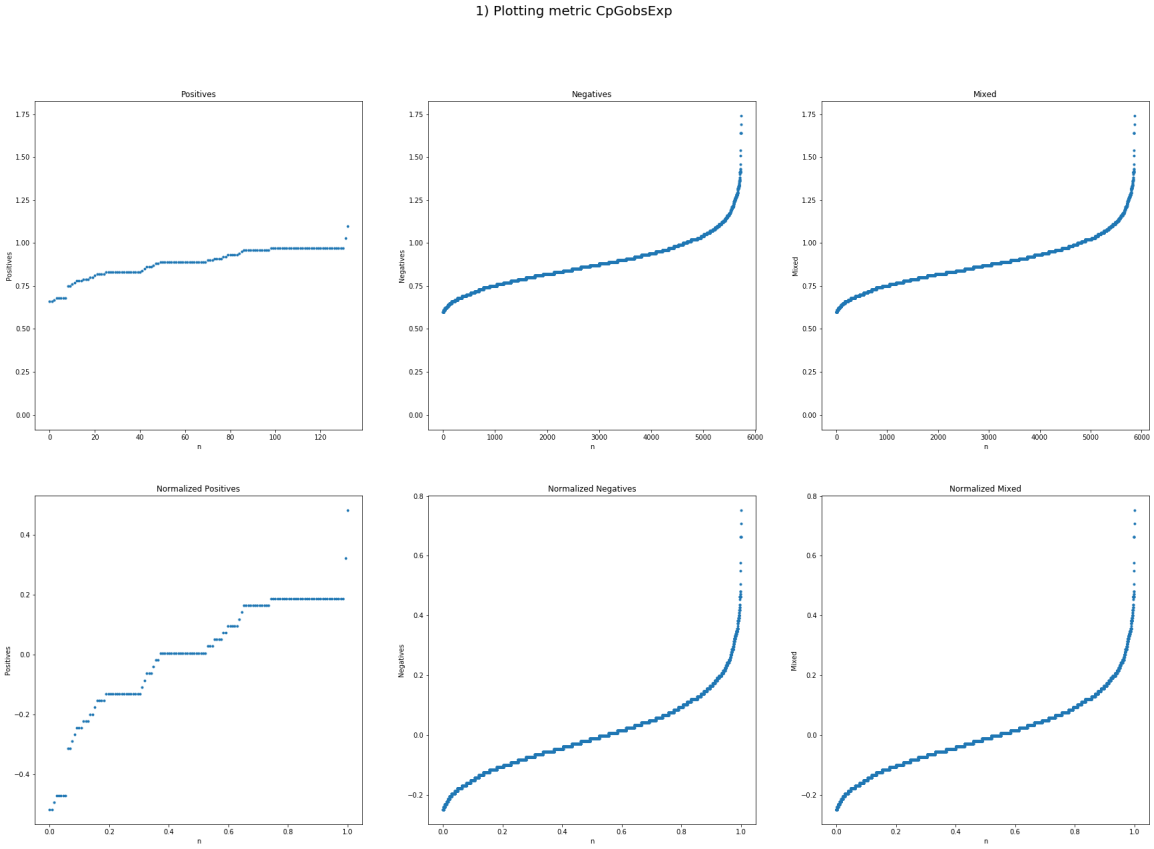


Figura 2.3: Values of metric CpGobsExp

2.3 CpGperCpG

2.3.1 Metric sample distribution

The data points seem to follow a **Beta** distribution.

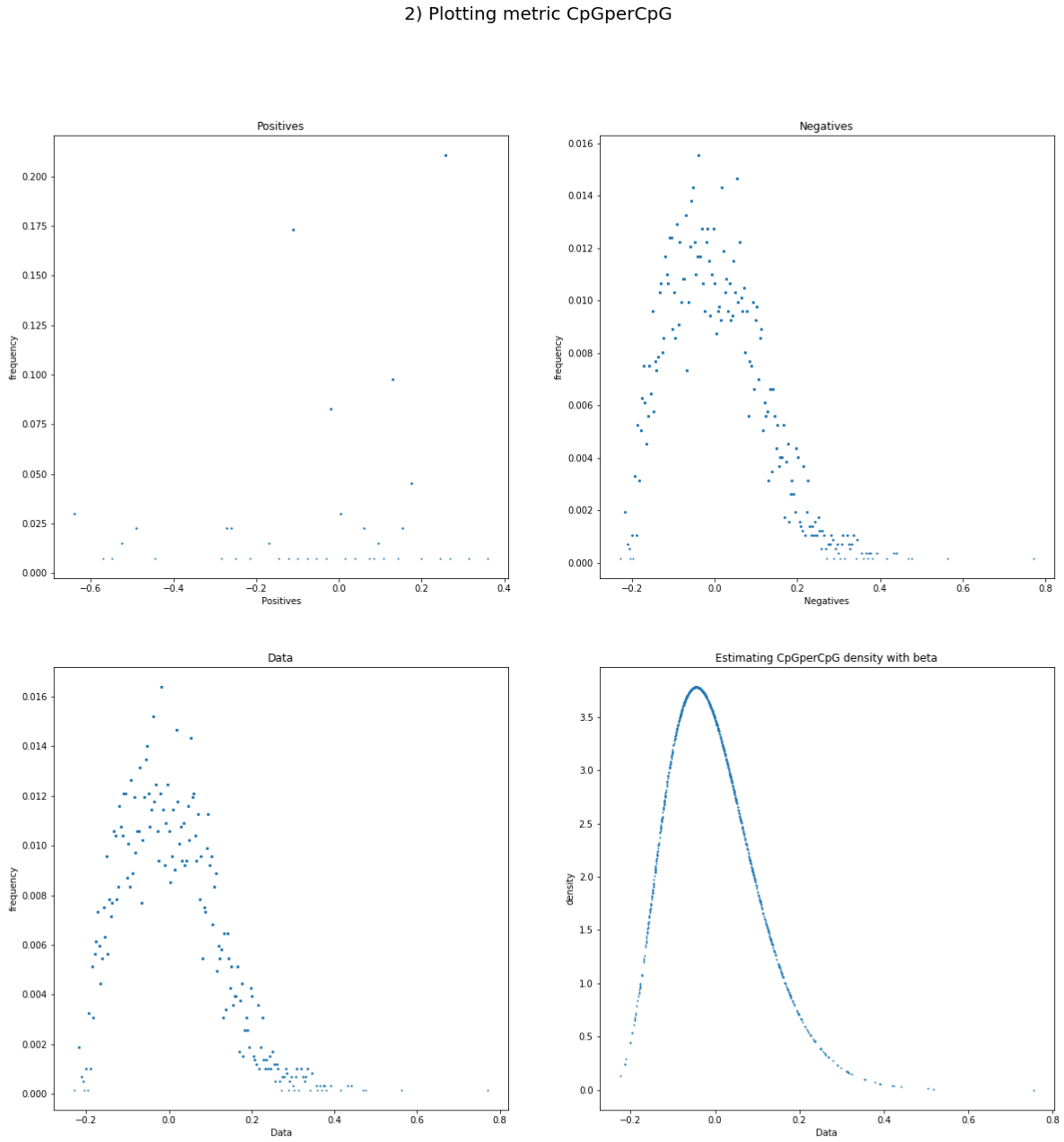


Figura 2.4: Sampling distribution of metric CpGperCpG

2.3.2 Metric values

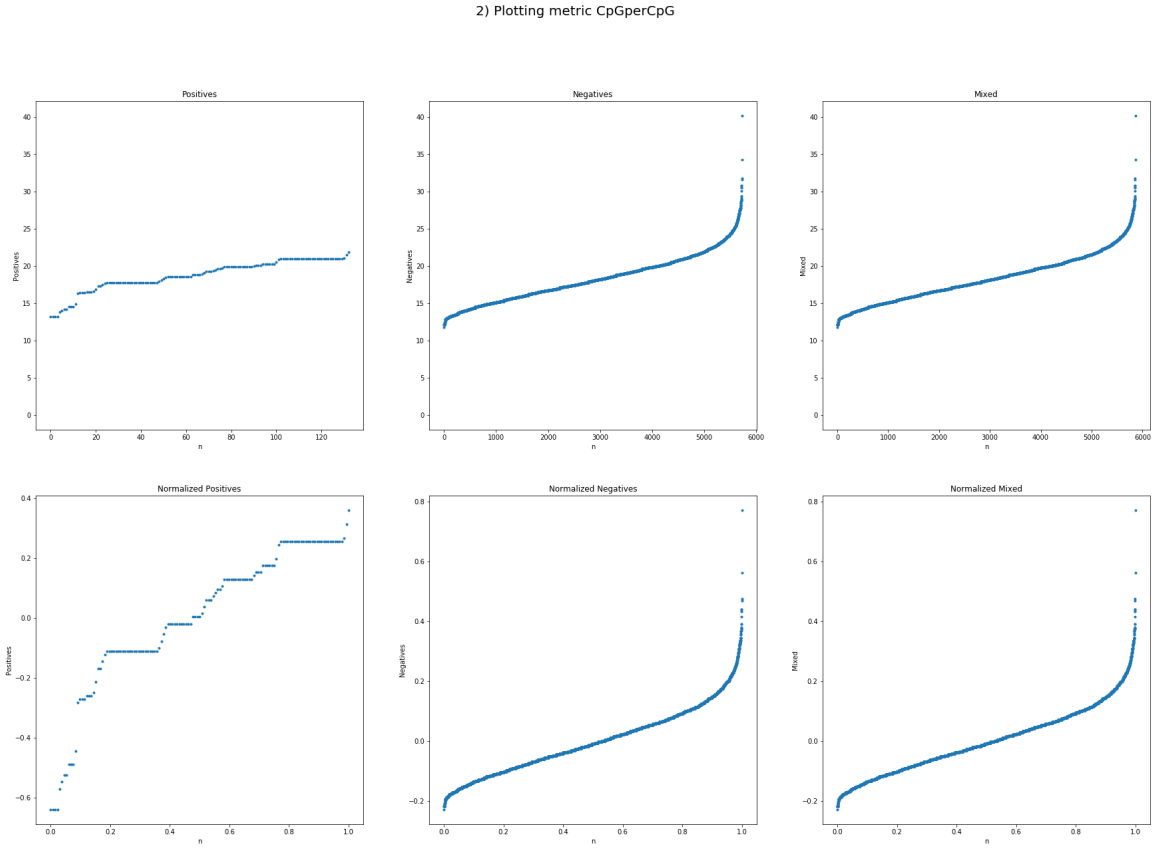


Figura 2.5: Values of metric CpGperCpG

2.4 CpGperGC

2.4.1 Metric sample distribution

The data points seem to follow a **Gaussian** distribution.

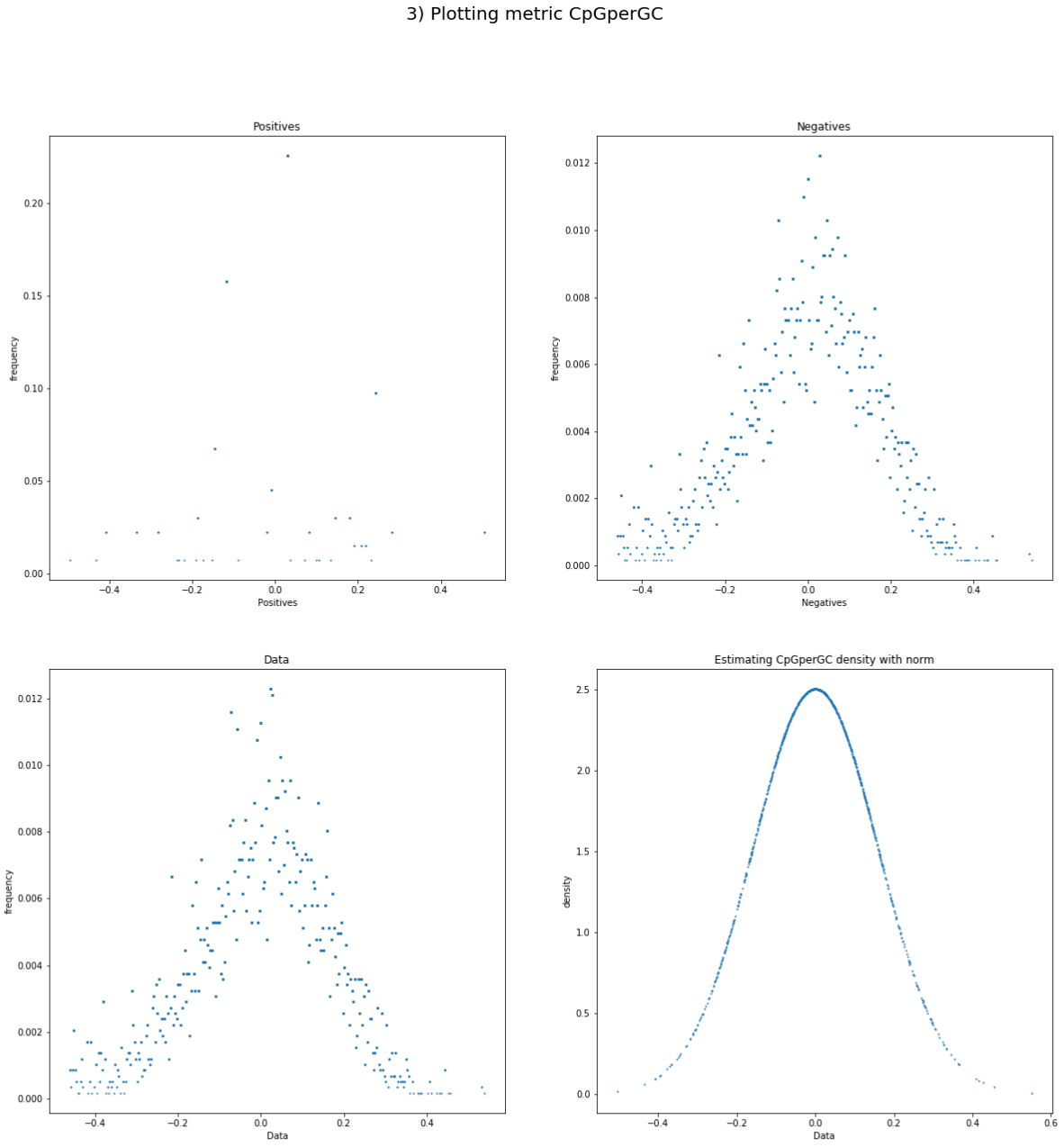


Figura 2.6: Sampling distribution of metric CpGperGC

2.4.2 Metric values

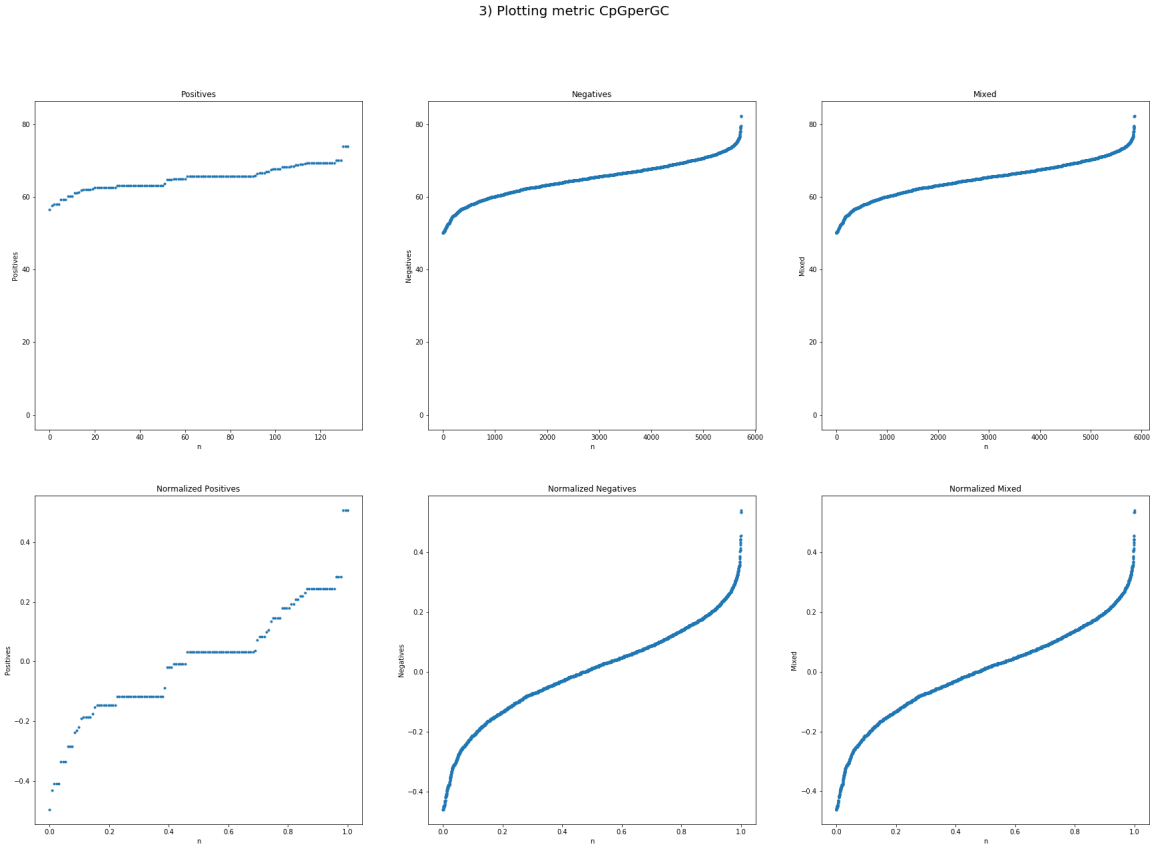


Figura 2.7: Values of metric CpGperGC

2.5 DGVCount

2.5.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

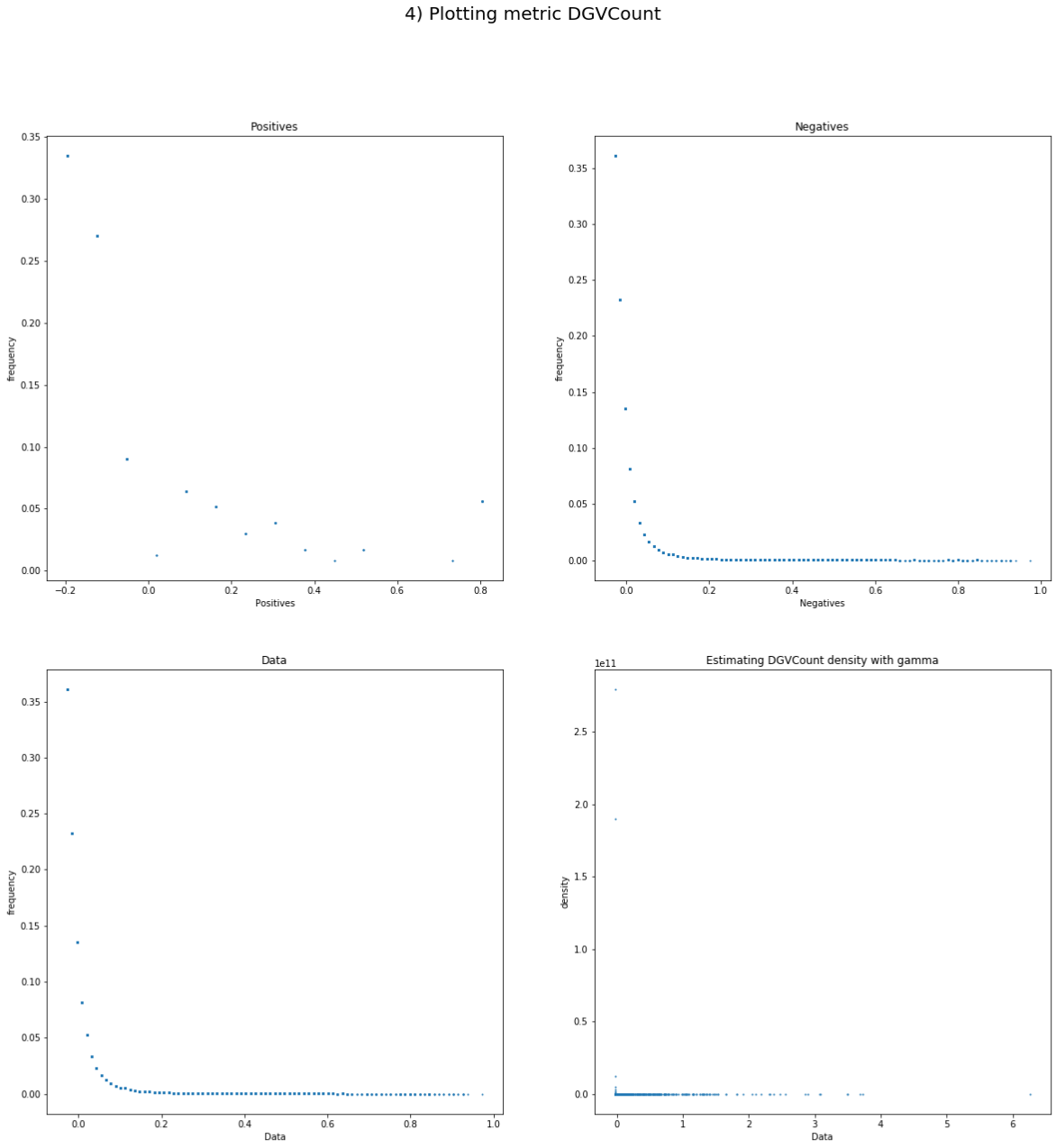


Figura 2.8: Sampling distribution of metric DGVCount

2.5.2 Metric values

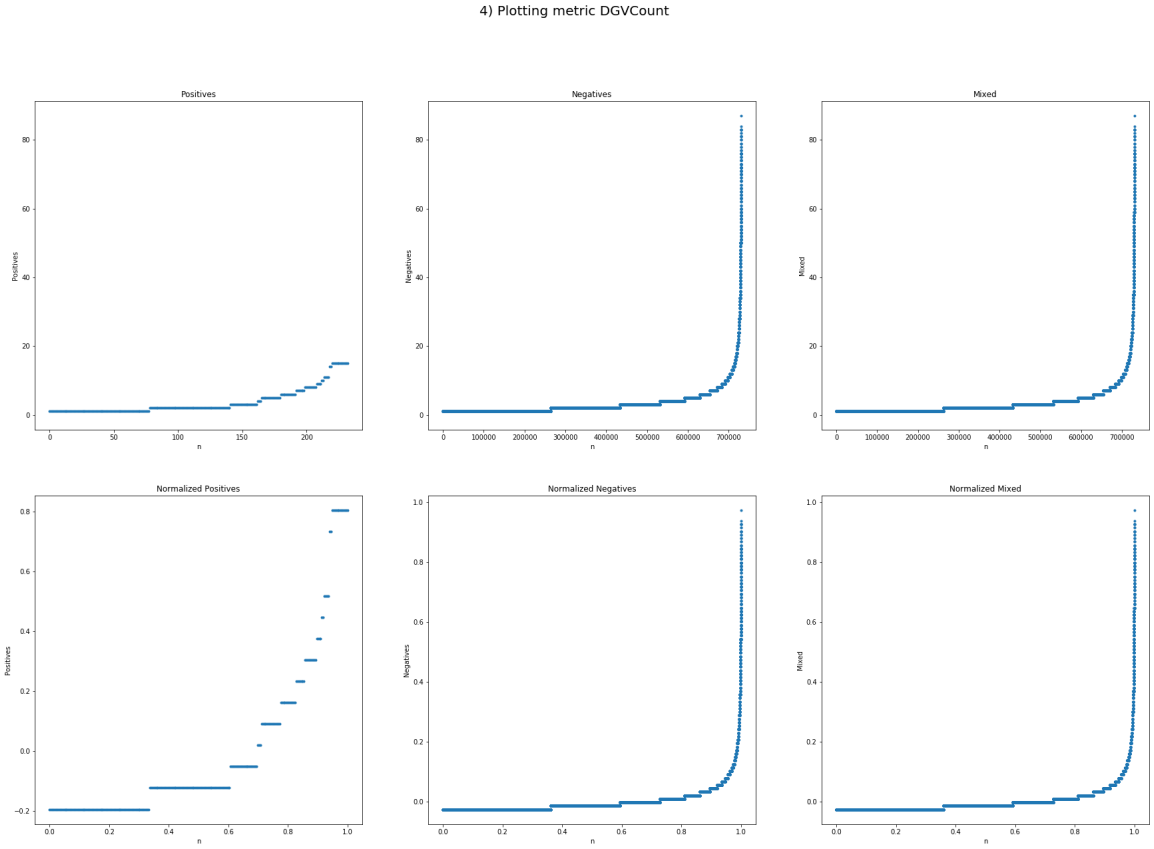


Figura 2.9: Values of metric DGVCount

2.6 DnaseClusteredHyp

2.6.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

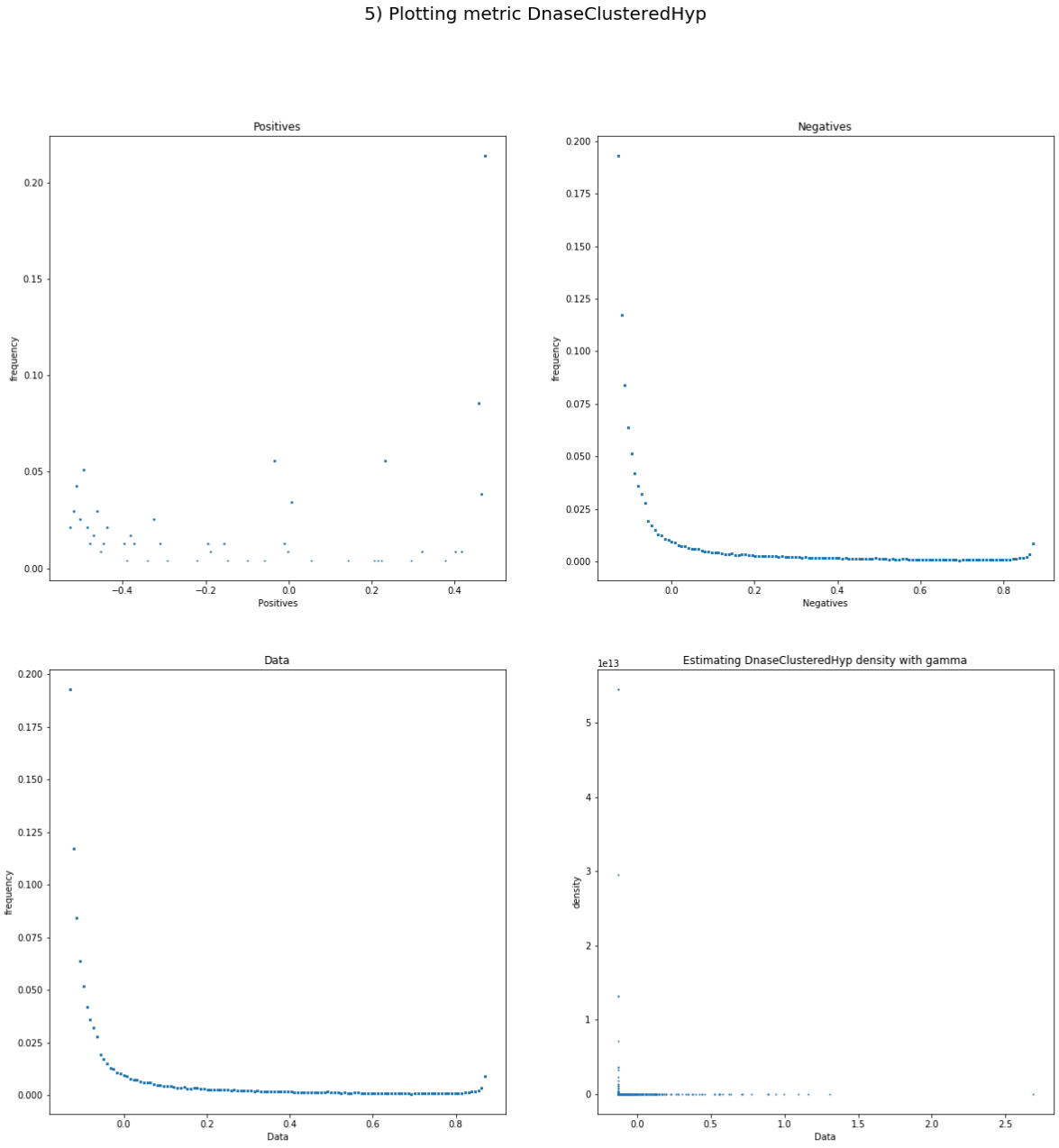


Figura 2.10: Sampling distribution of metric DnaseClusteredHyp

2.6.2 Metric values

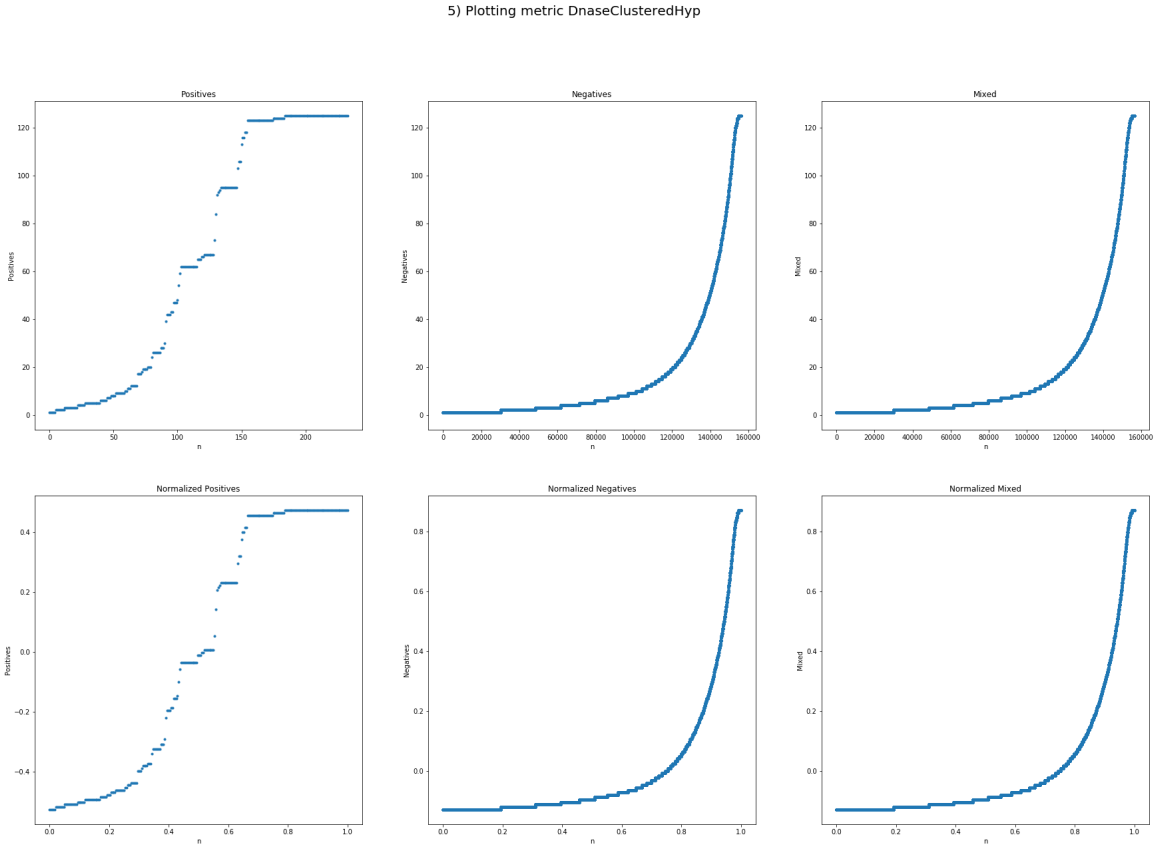


Figura 2.11: Values of metric DnaseClusteredHyp

2.7 DnaseClusteredScore

2.7.1 Metric sample distribution

The data points seem to follow **slightly** a **Beta** distribution.

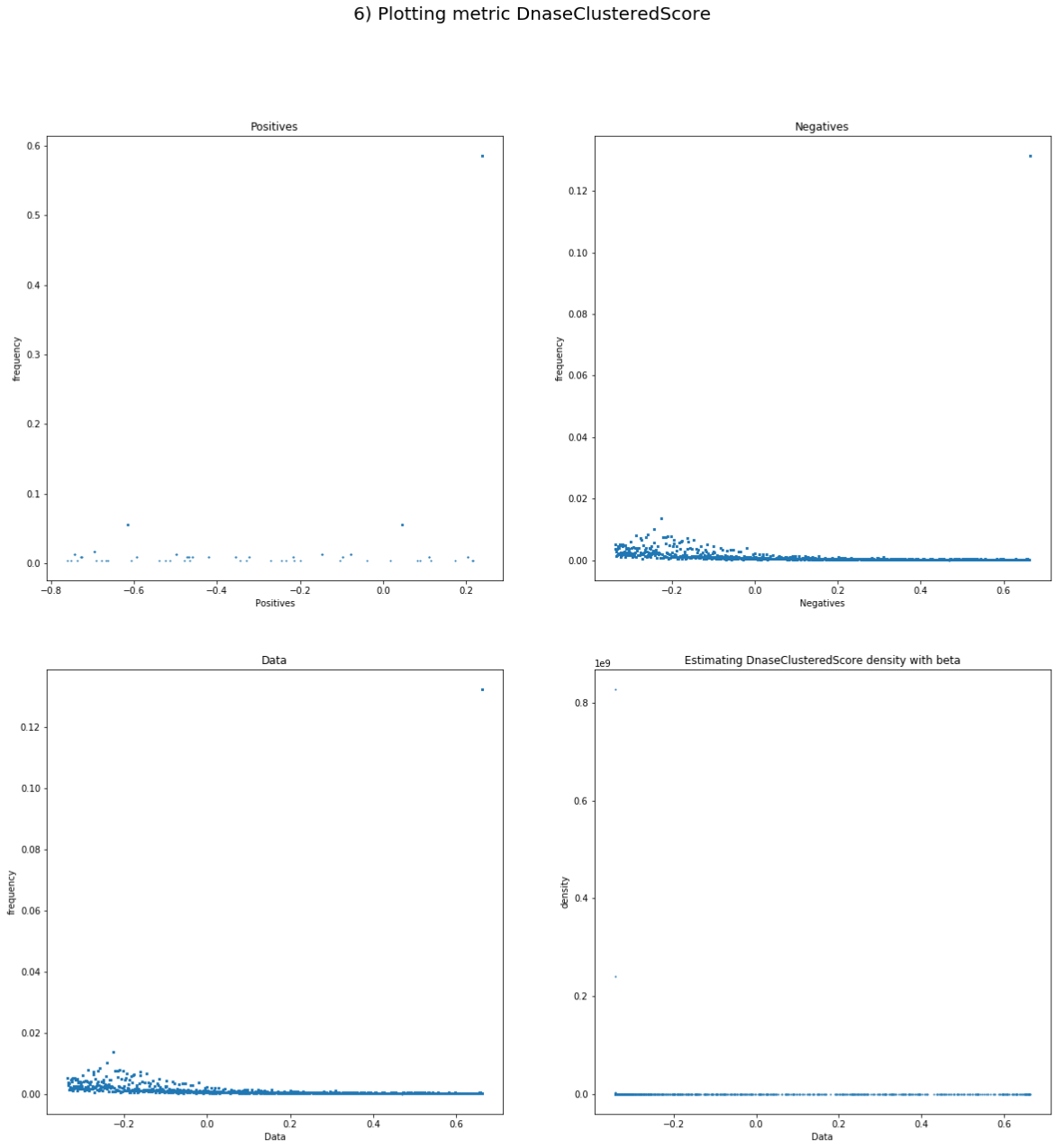


Figura 2.12: Sampling distribution of metric DnaseClusteredScore

2.7.2 Metric values

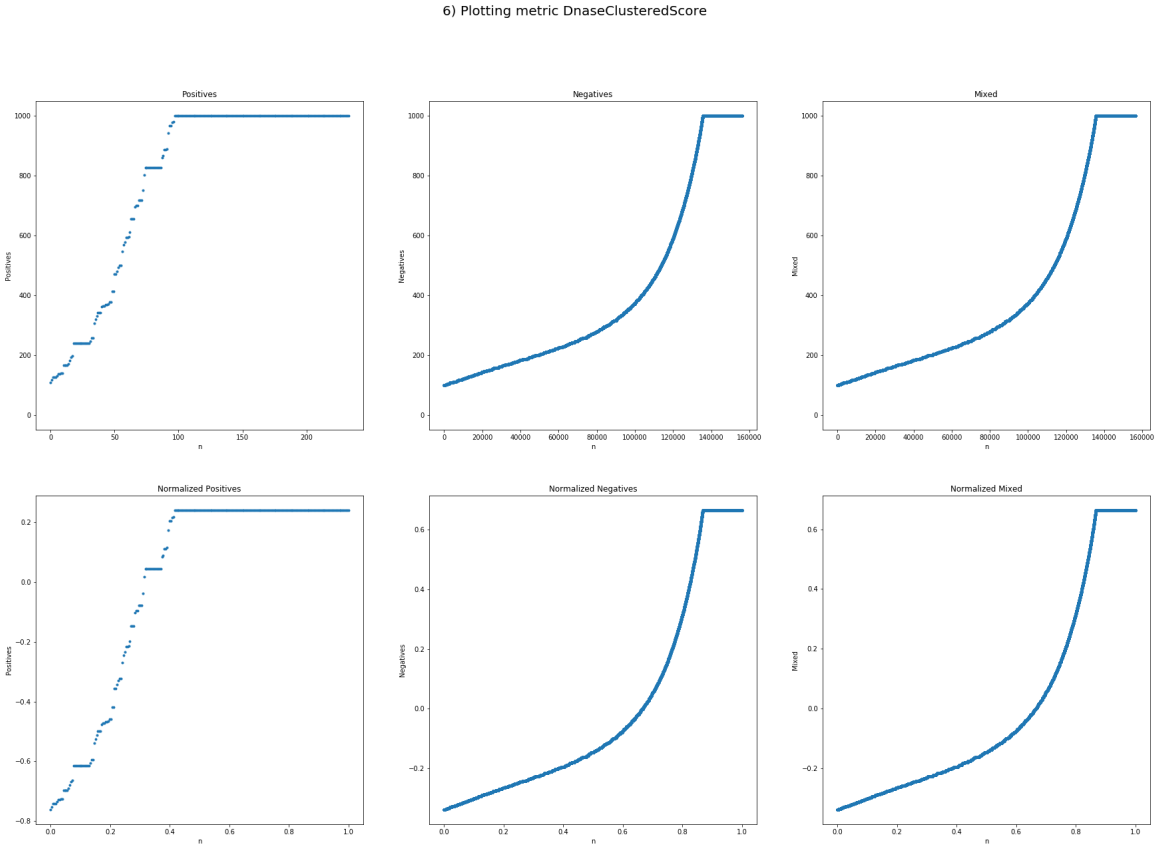


Figura 2.13: Values of metric DnaseClusteredScore

2.8 EncH3K27Ac

2.8.1 Metric sample distribution

The data points seem to follow a family of **Gamma** distributions (a speculation for this distribution could be the different groups from which the data are extracted), we will approximate them to one with a linear combination of the parameters.

7) Plotting metric EncH3K27Ac

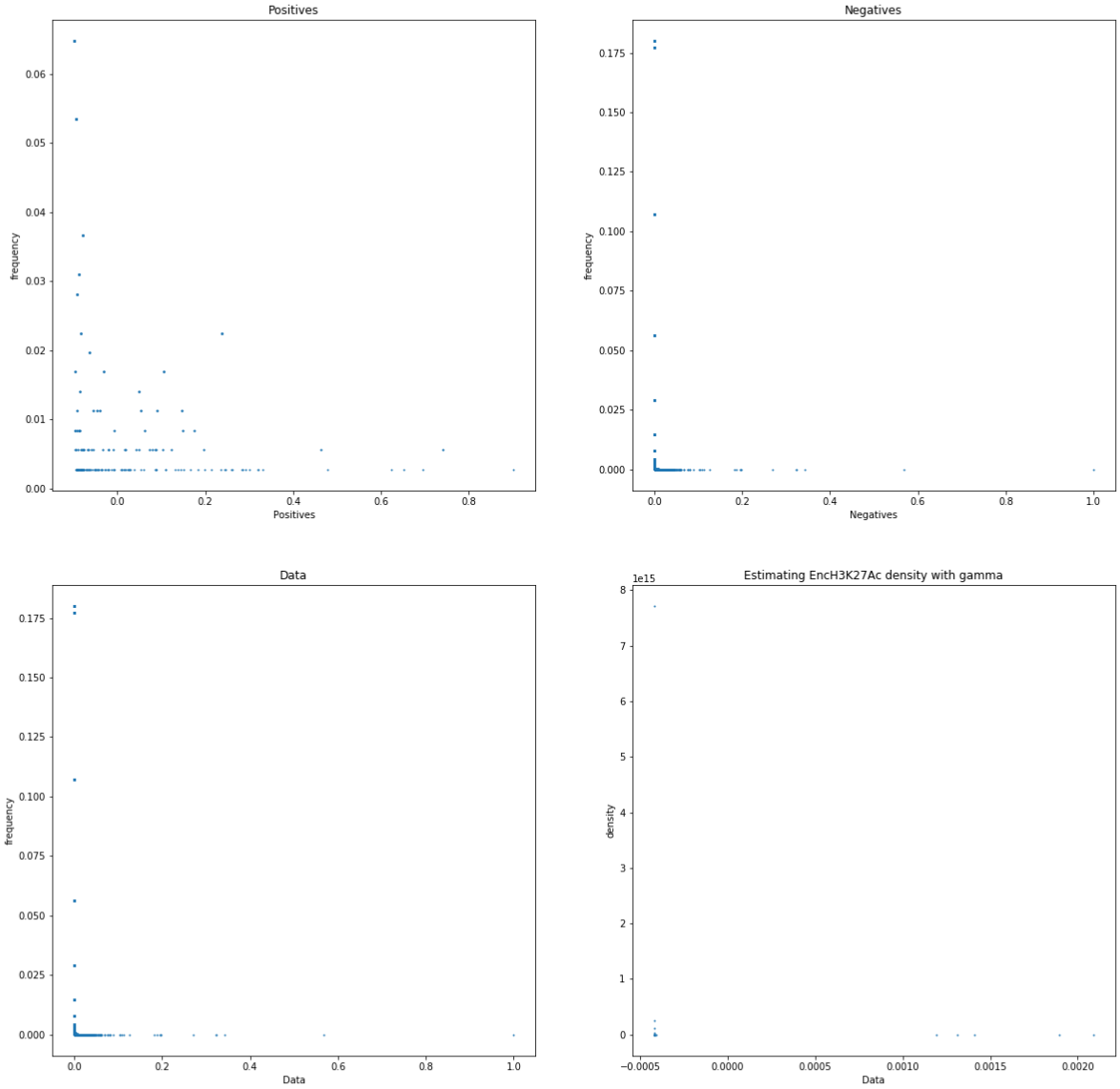


Figura 2.14: Sampling distribution of metric EncH3K27Ac

2.8.2 Metric values

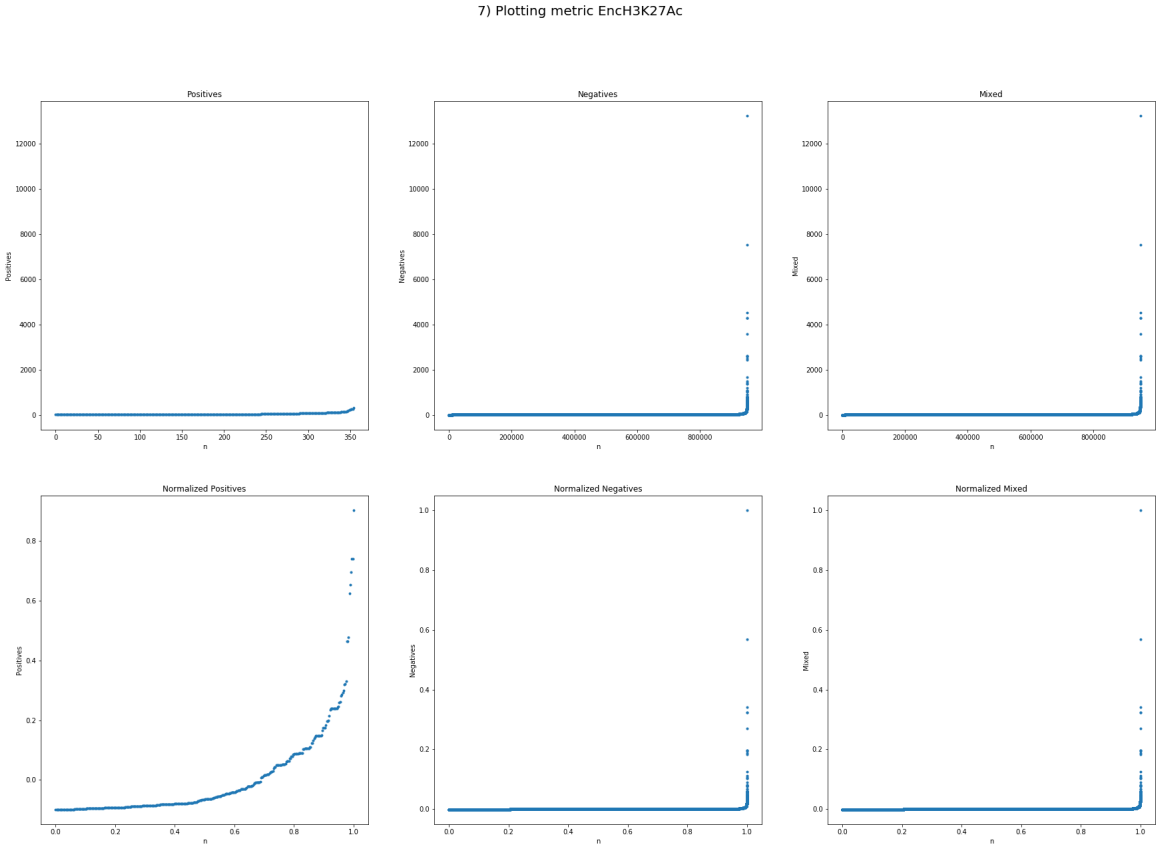


Figura 2.15: Values of metric EncH3K27Ac

2.9 EncH3K4Me1

2.9.1 Metric sample distribution

The data points seem to follow a family of **Gamma** distributions (a speculation for this distribution could be the different groups from which the data are extracted), we will approximate them to one with a linear combination of the parameters.

8) Plotting metric EncH3K4Me1

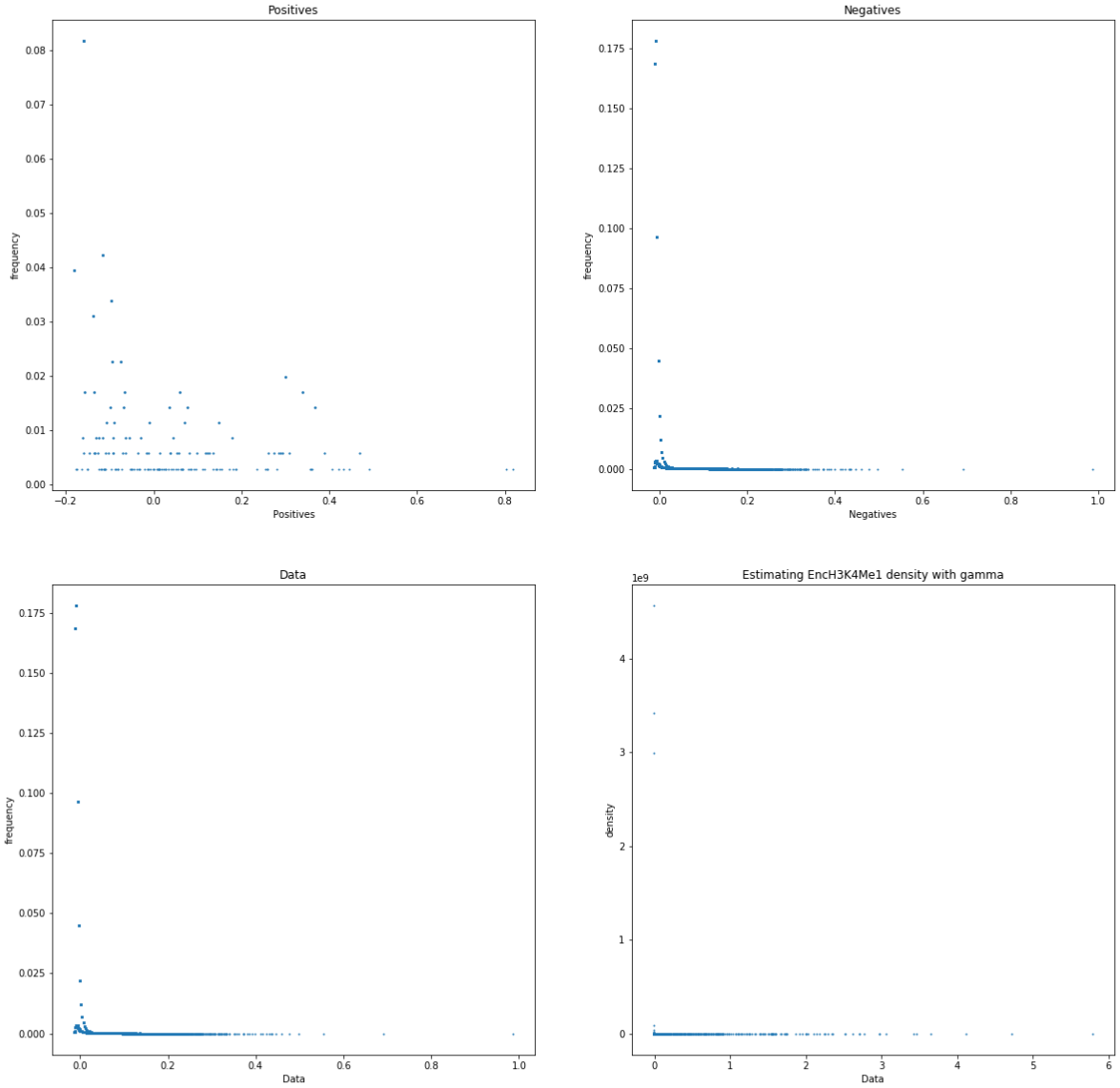


Figura 2.16: Sampling distribution of metric EncH3K4Me1

2.9.2 Metric values

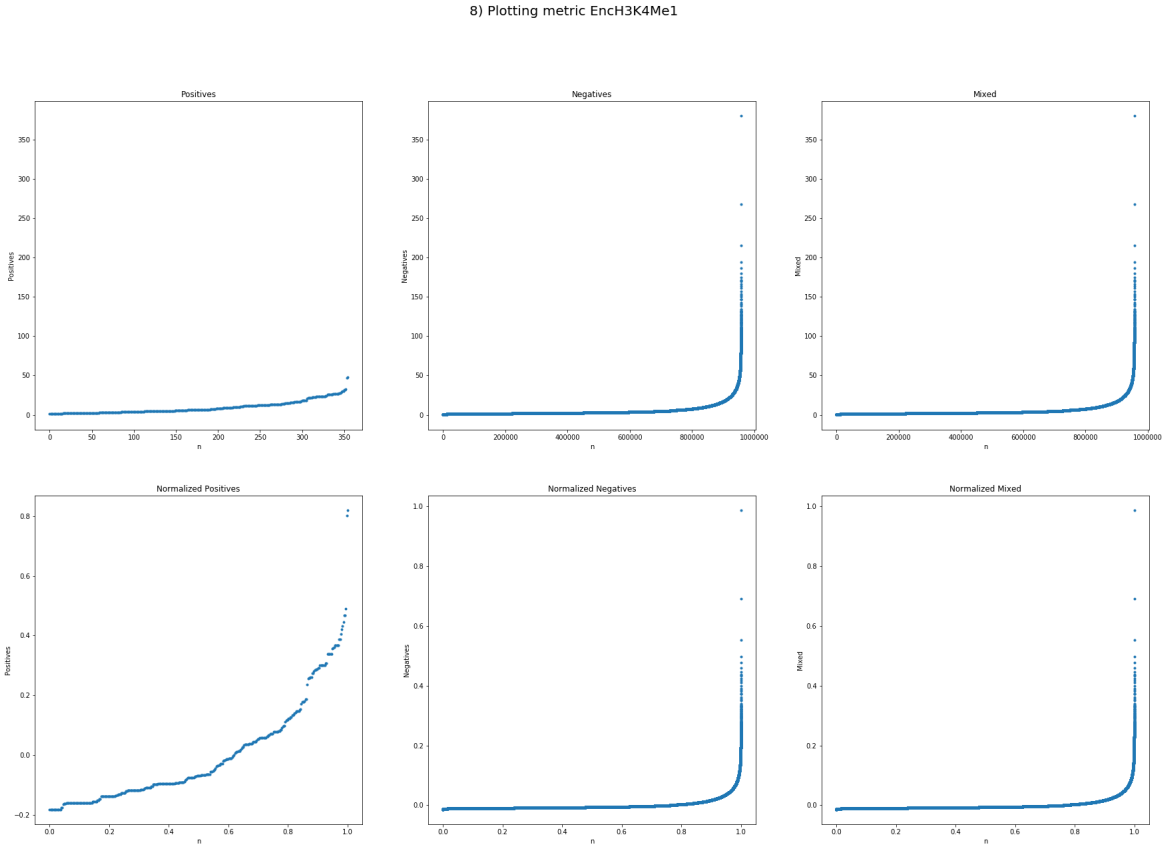


Figura 2.17: Values of metric EncH3K4Me1

2.10 EncH3K4Me3

2.10.1 Metric sample distribution

The data points seem to follow a family of **Gamma** distributions (a speculation for this distribution could be the different groups from which the data are extracted), we will approximate them to one with a linear combination of the parameters.

9) Plotting metric EncH3K4Me3

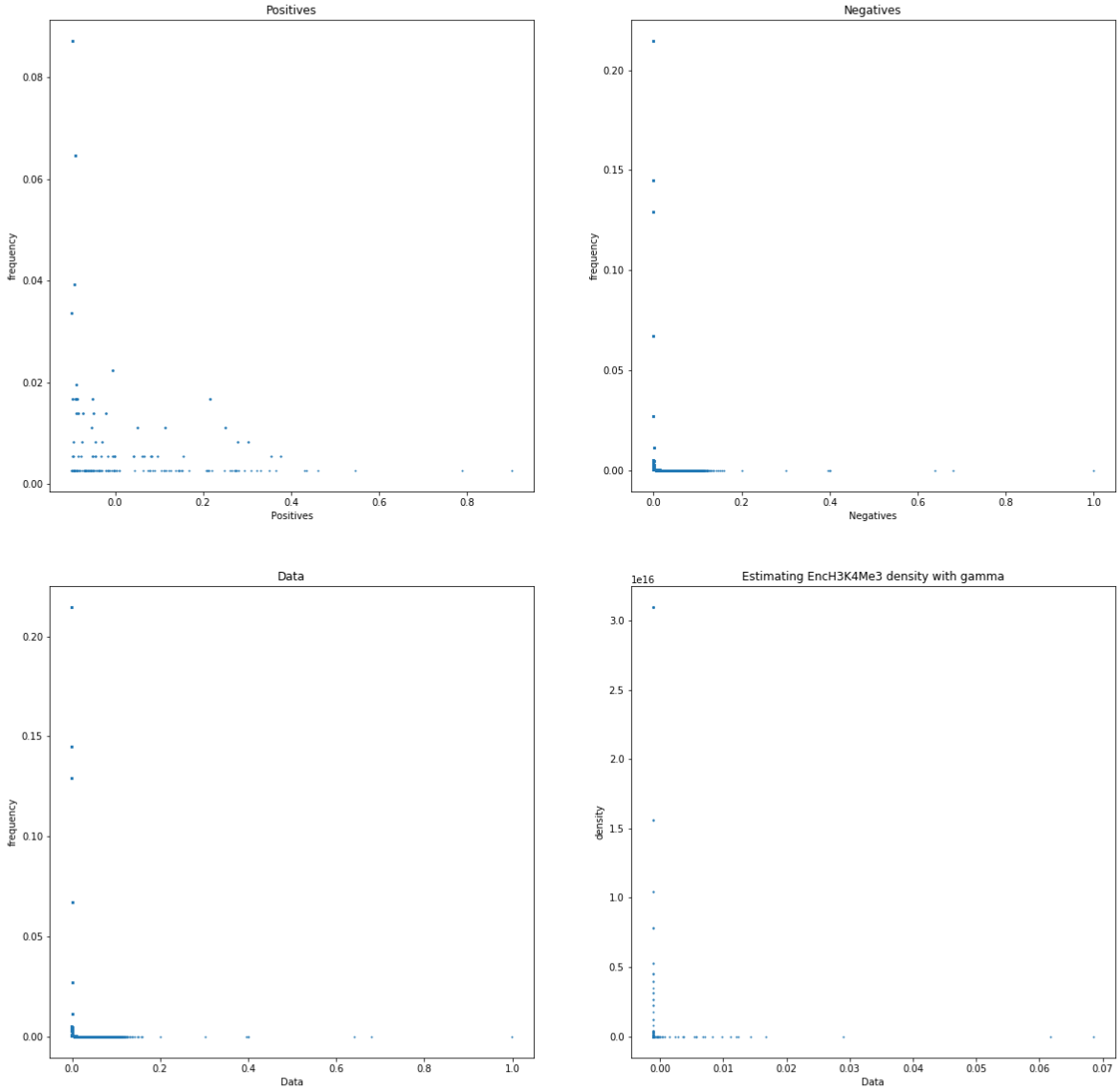


Figura 2.18: Sampling distribution of metric EncH3K4Me3

2.10.2 Metric values

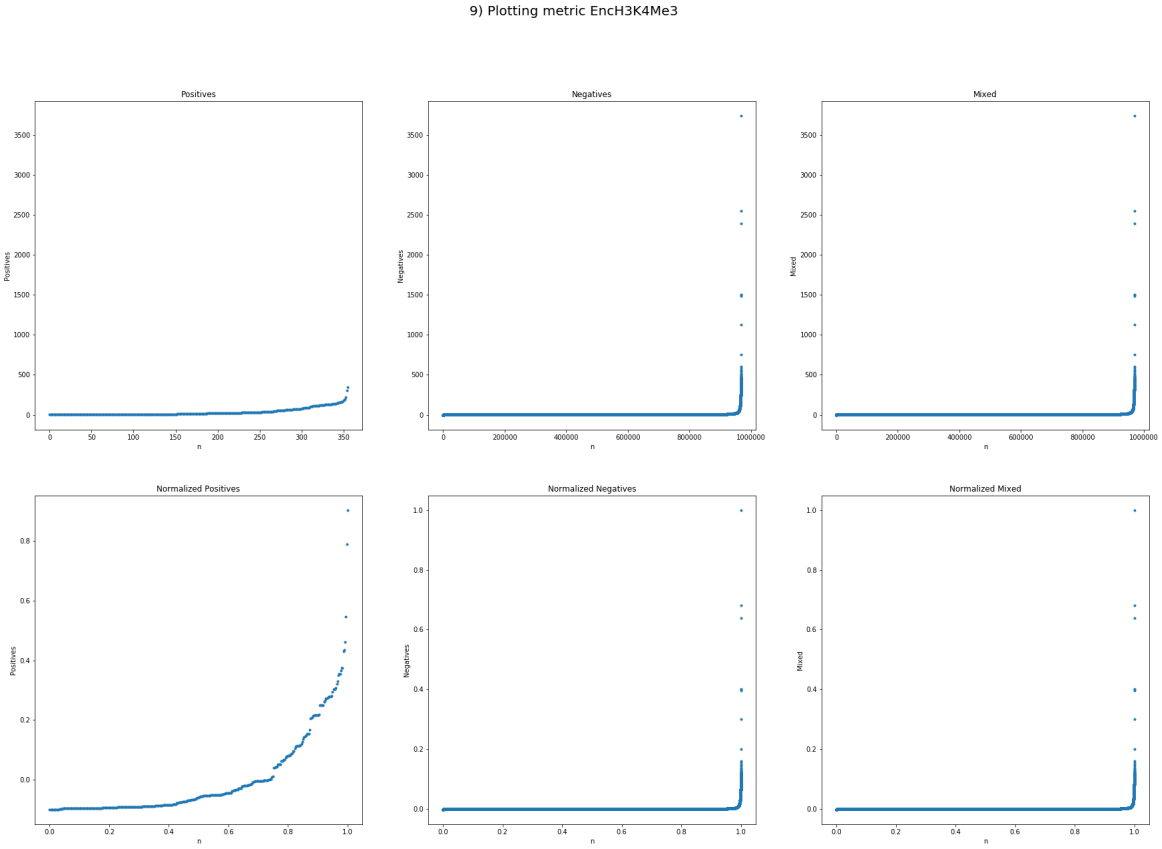


Figura 2.19: Values of metric EncH3K4Me3

2.11 GCContent

2.11.1 Metric sample distribution

The data points seem to be a combination of two **Gaussian** distributions.

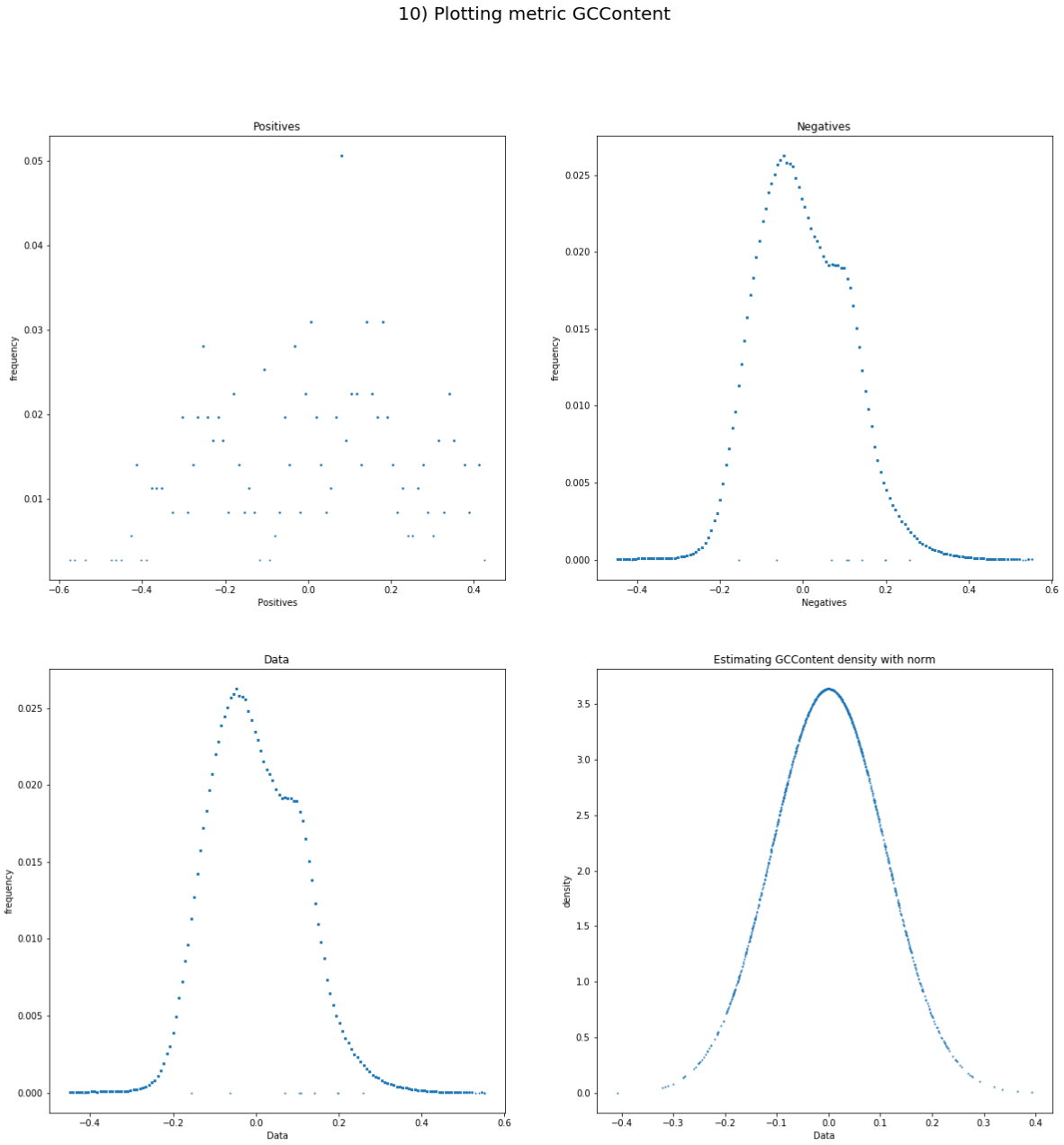


Figura 2.20: Sampling distribution of metric GCContent

2.11.2 Metric values

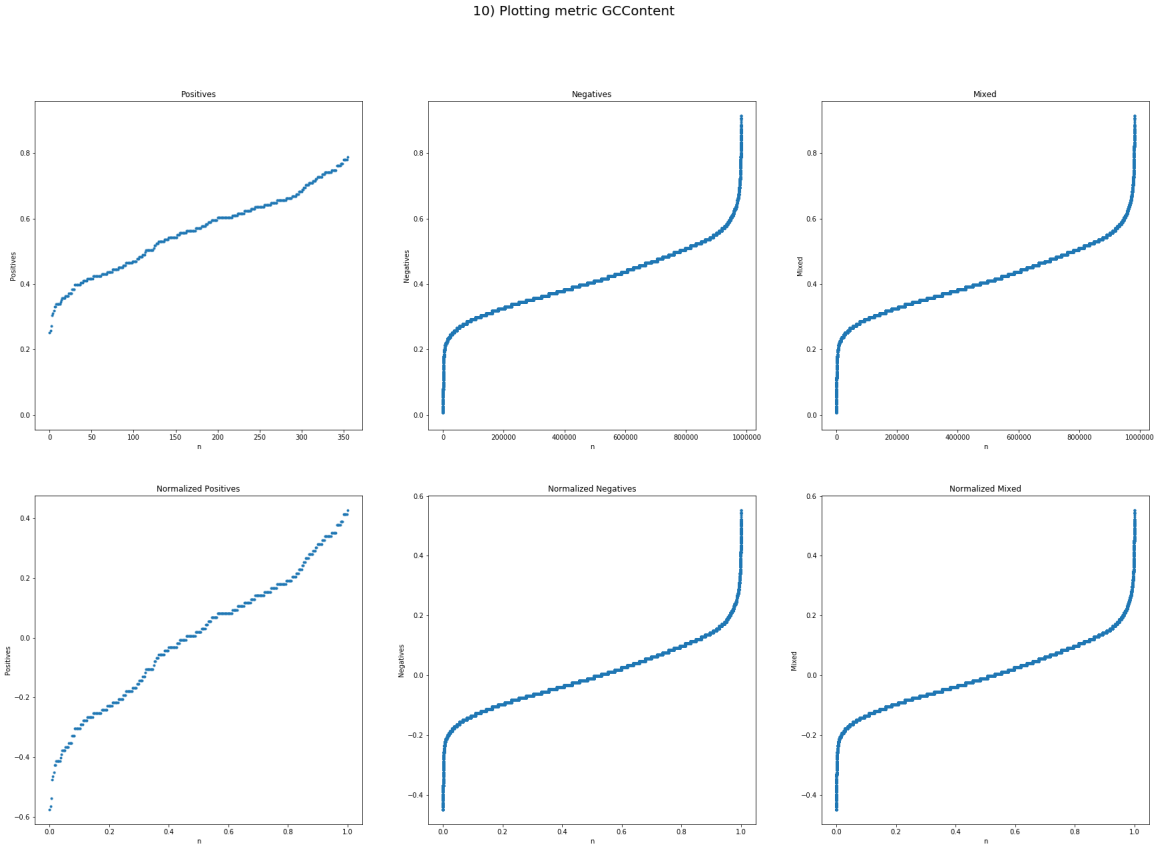


Figura 2.21: Values of metric GCContent

2.12 GerpRS

2.12.1 Metric sample distribution

The data points seem to follow a family of **Gamma** distributions (a speculation for this distribution could be the different groups from which the data are extracted), we will approximate them to one with a linear combination of the parameters.

11) Plotting metric GerpRS

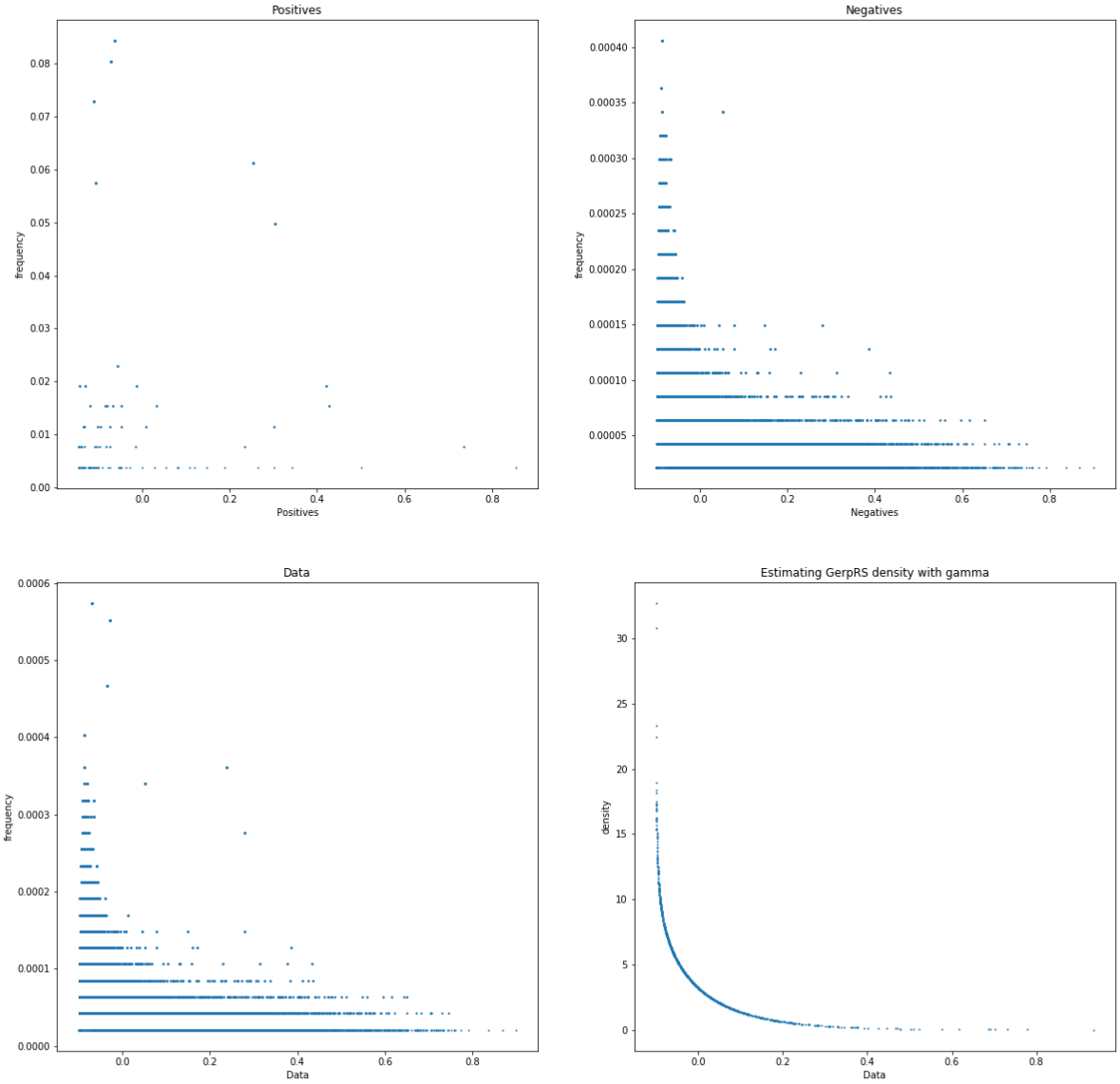


Figura 2.22: Sampling distribution of metric GerpRS

2.12.2 Metric values

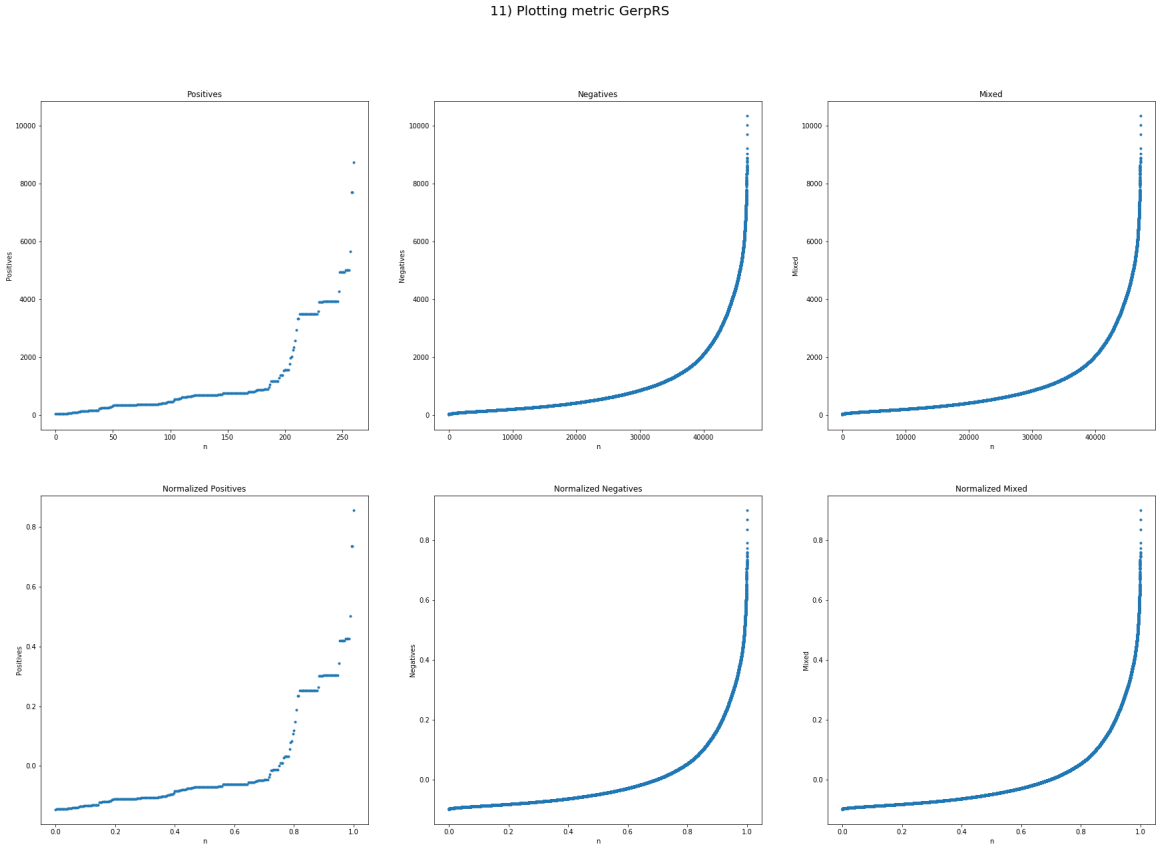


Figura 2.23: Values of metric GerpRS

2.13 GerpRSpv

2.13.1 Metric sample distribution

The data points seem to follow a family of **Gamma** distributions (a speculation for this distribution could be the different groups from which the data are extracted), we will approximate them to one with a linear combination of the parameters.

12) Plotting metric GerpRSpv

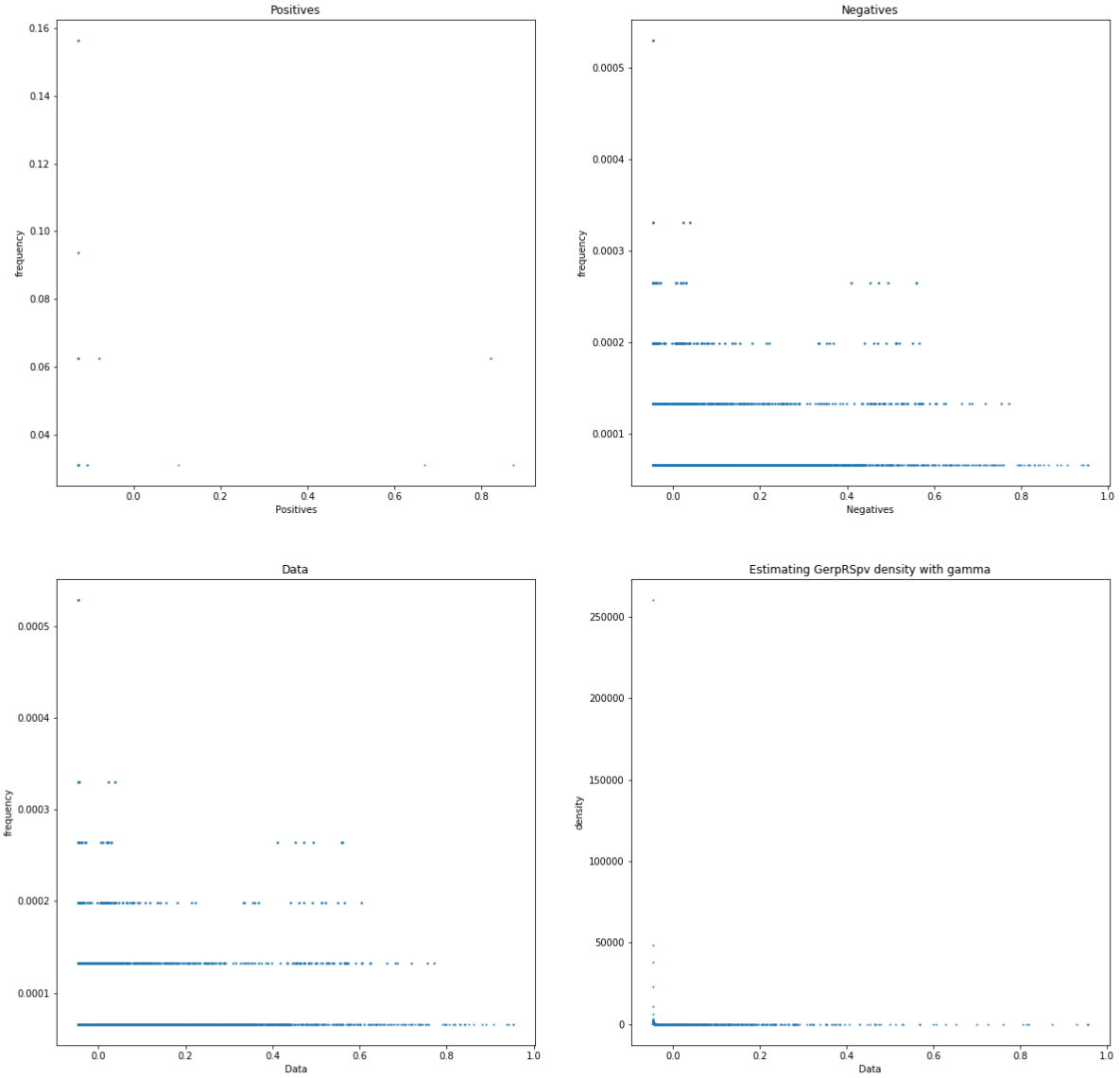


Figura 2.24: Sampling distribution of metric GerpRSpv

2.13.2 Metric values

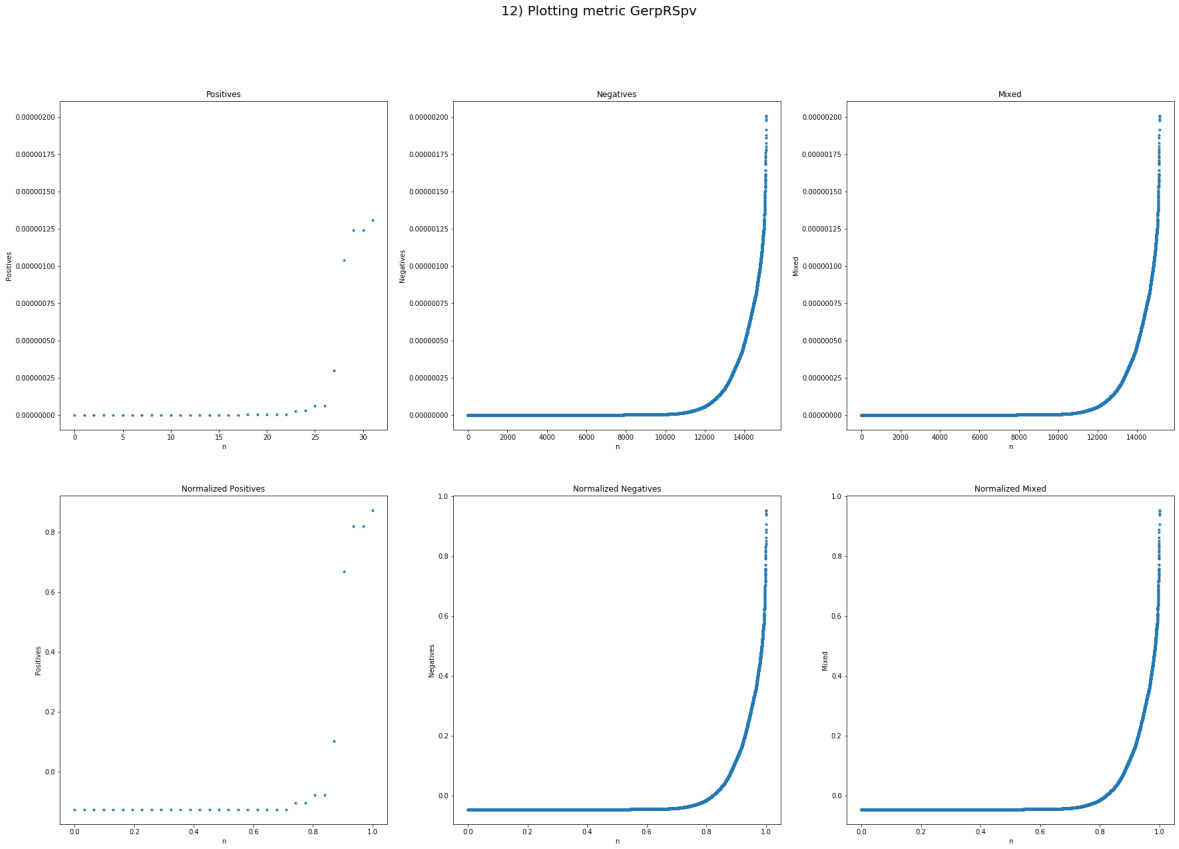


Figura 2.25: Values of metric GerpRSpv

2.14 ISCApath

2.14.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

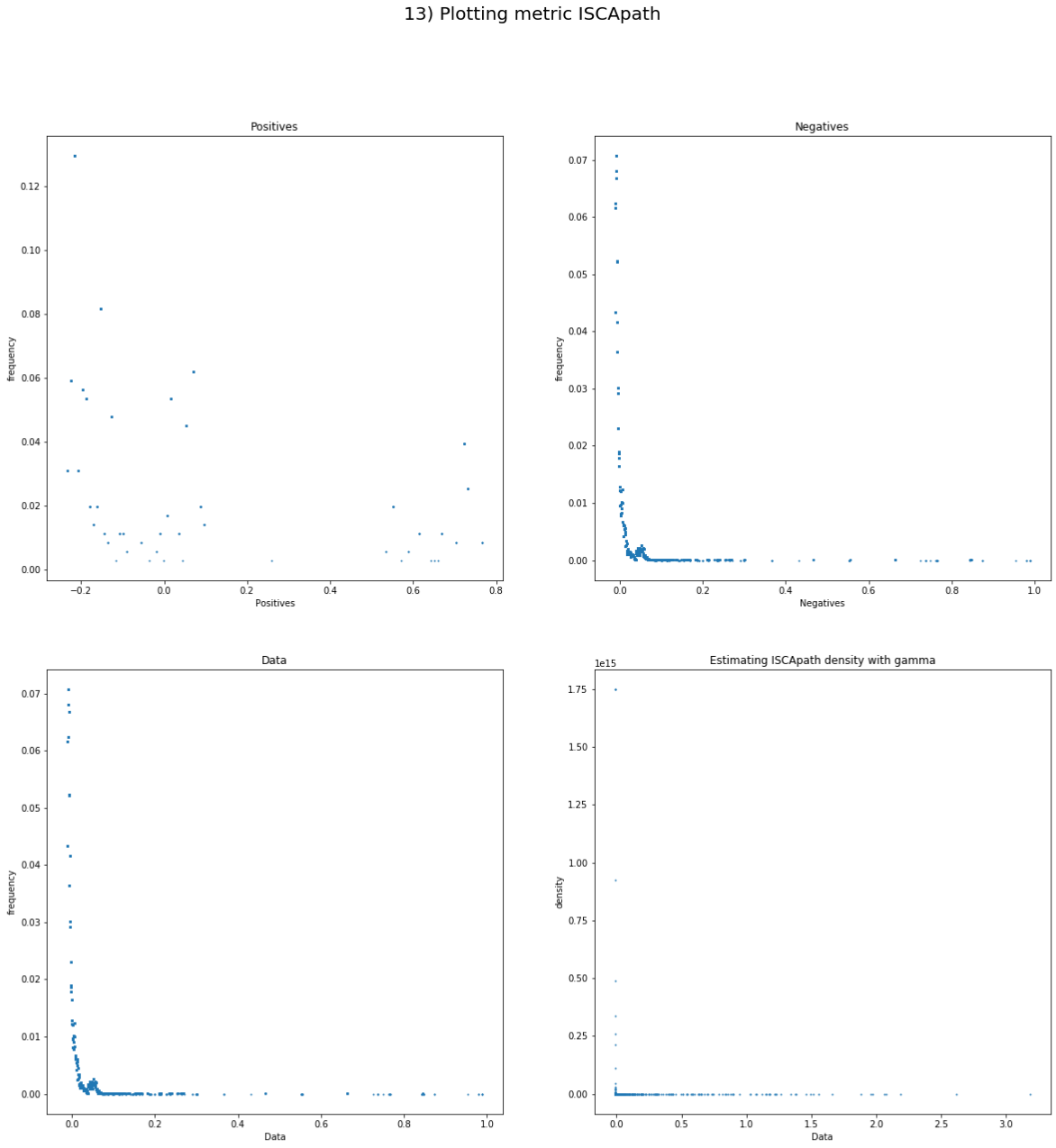


Figura 2.26: Sampling distribution of metric ISCApath

2.14.2 Metric values

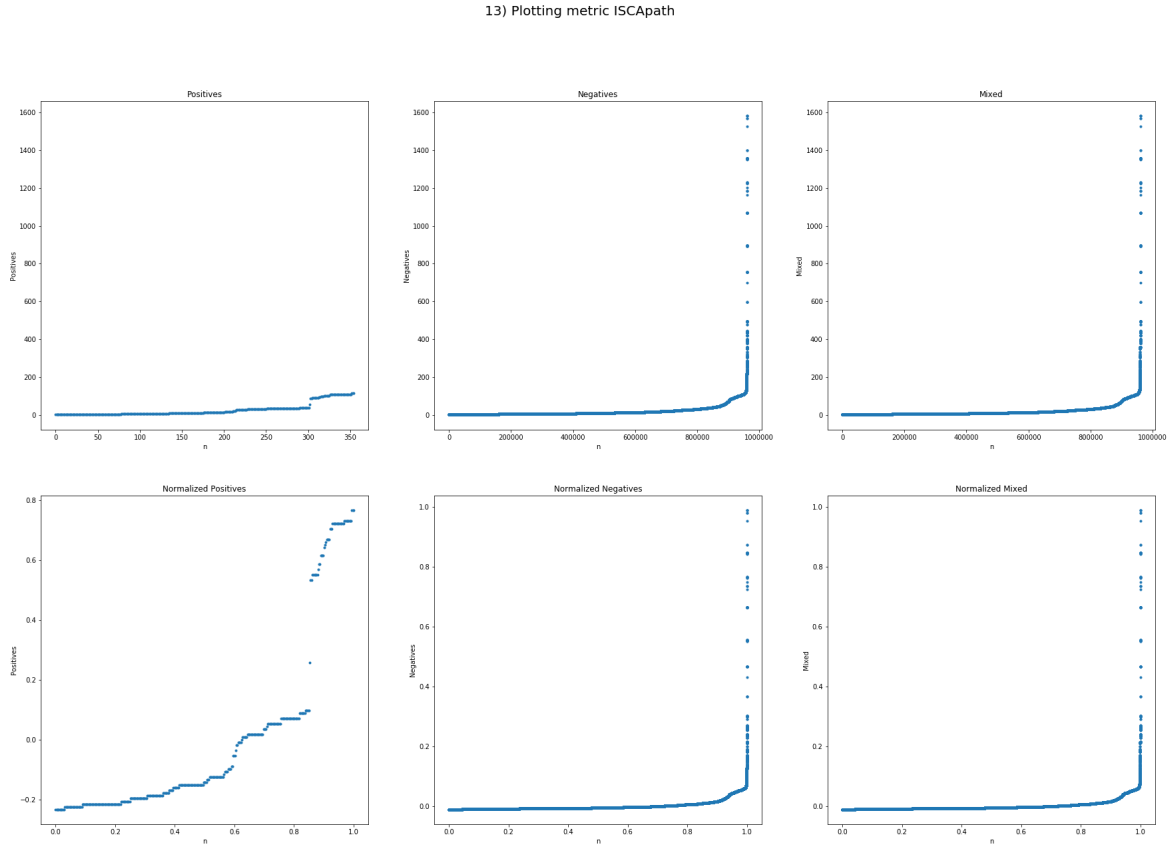


Figura 2.27: Values of metric ISCApath

2.15 commonVar

2.15.1 Metric sample distribution

The data points seem to follow an **Exponential Weibull** distribution.

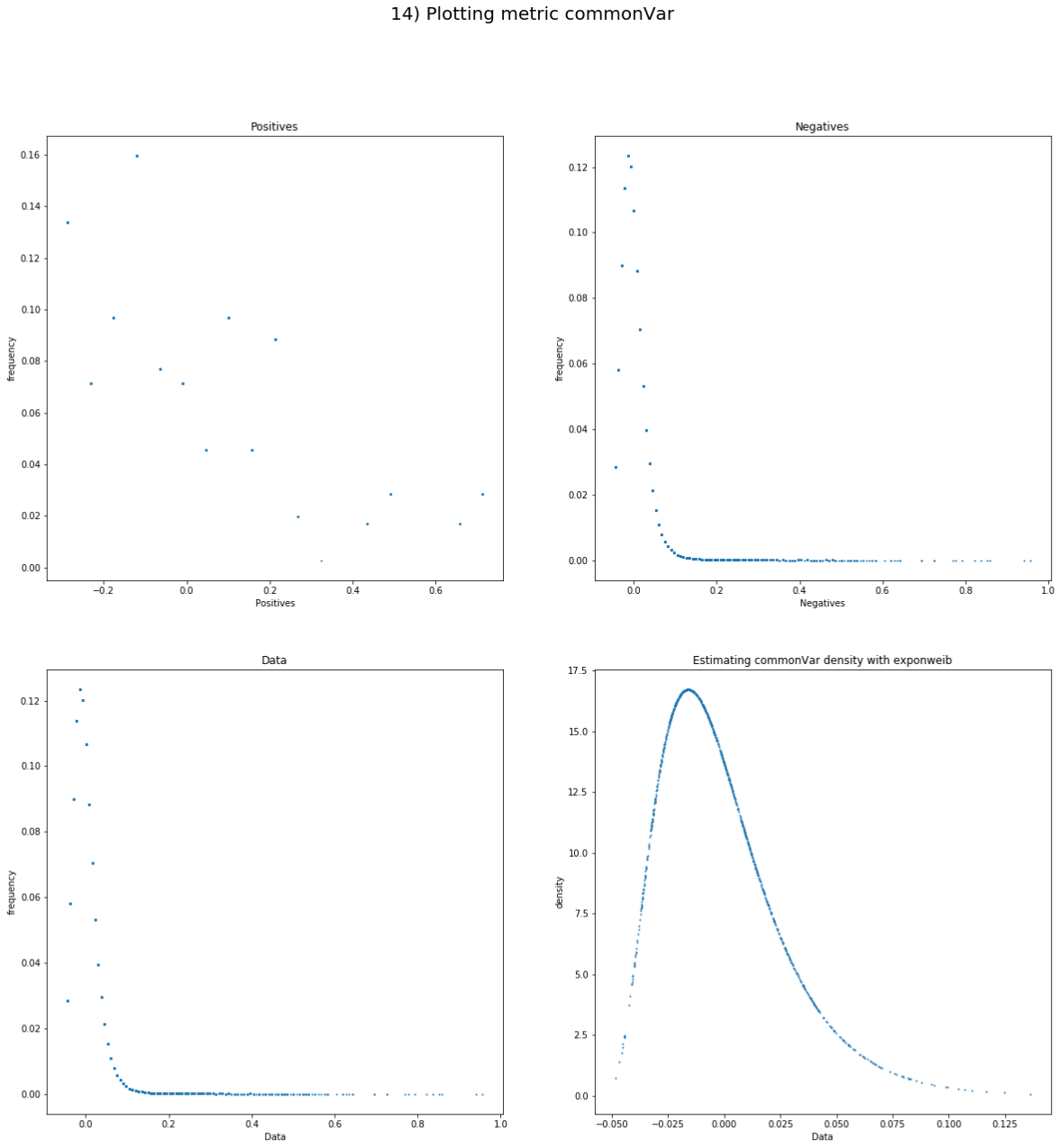


Figura 2.28: Sampling distribution of metric commonVar

2.15.2 Metric values

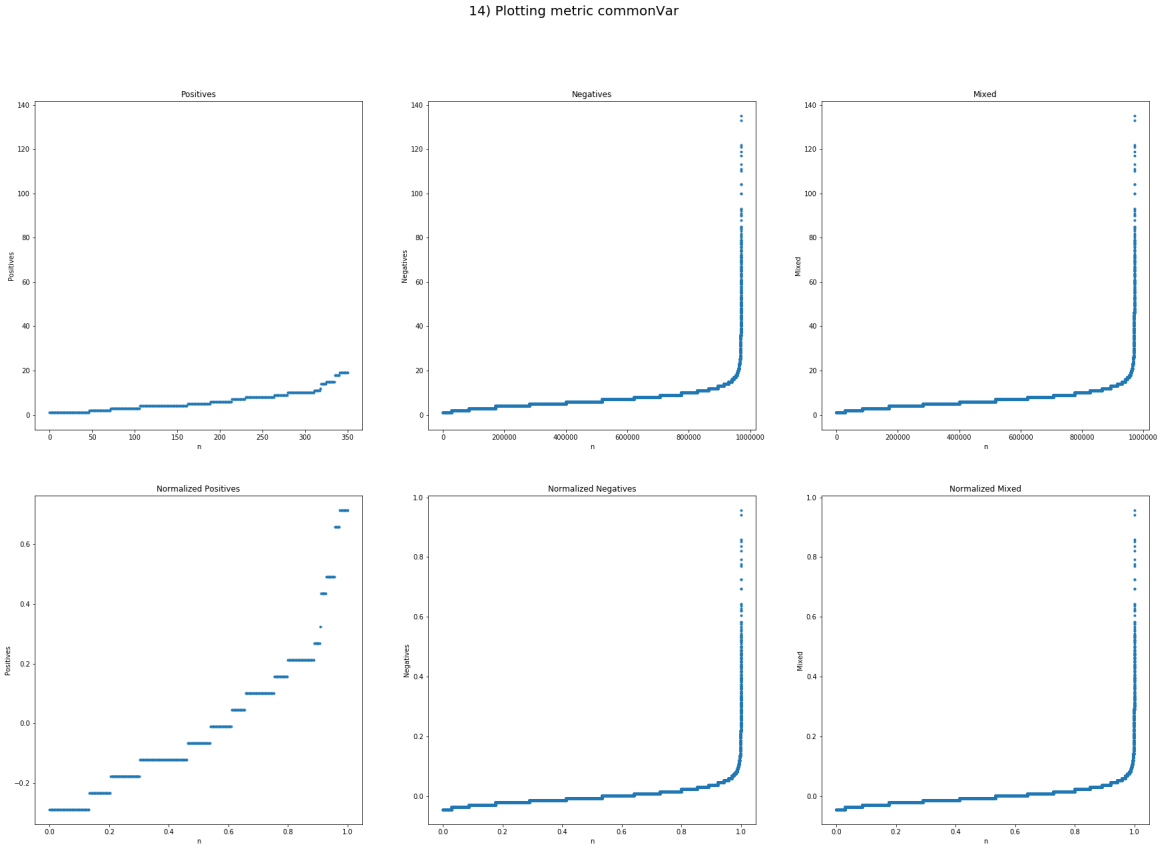


Figura 2.29: Values of metric commonVar

2.16 dbVARCount

2.16.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

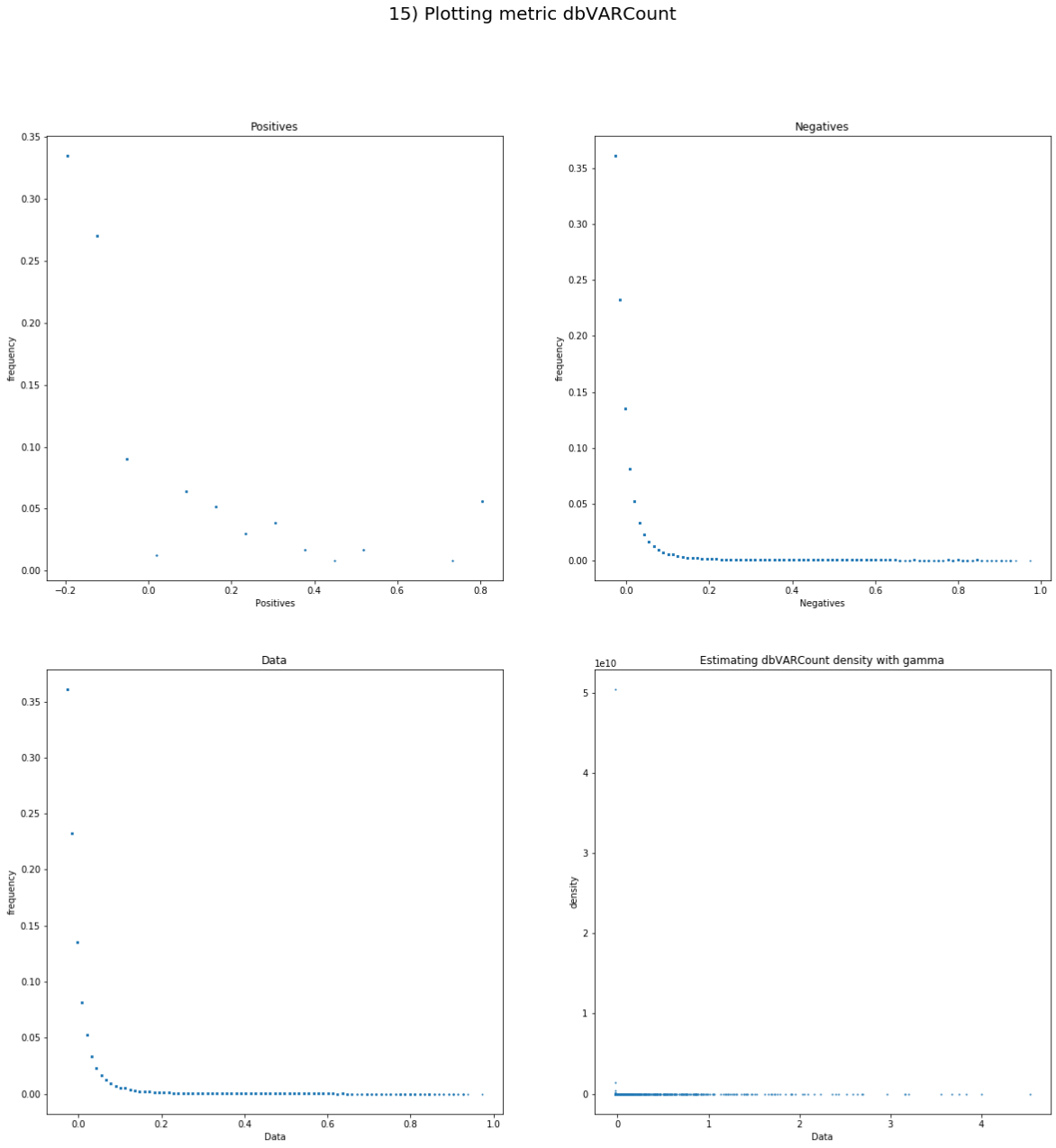


Figura 2.30: Sampling distribution of metric dbVARCount

2.16.2 Metric values

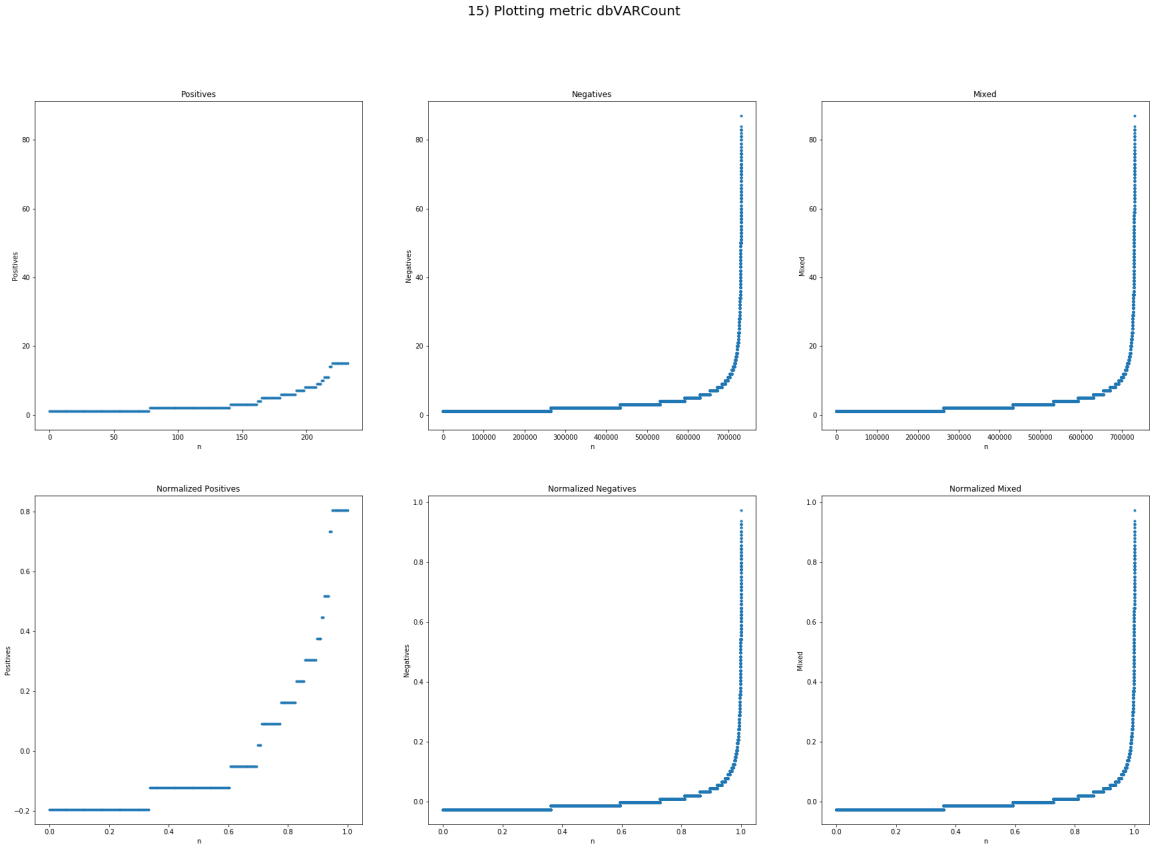


Figura 2.31: Values of metric dbVARCount

2.17 fantom5Perm

2.17.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

16) Plotting metric fantom5Perm

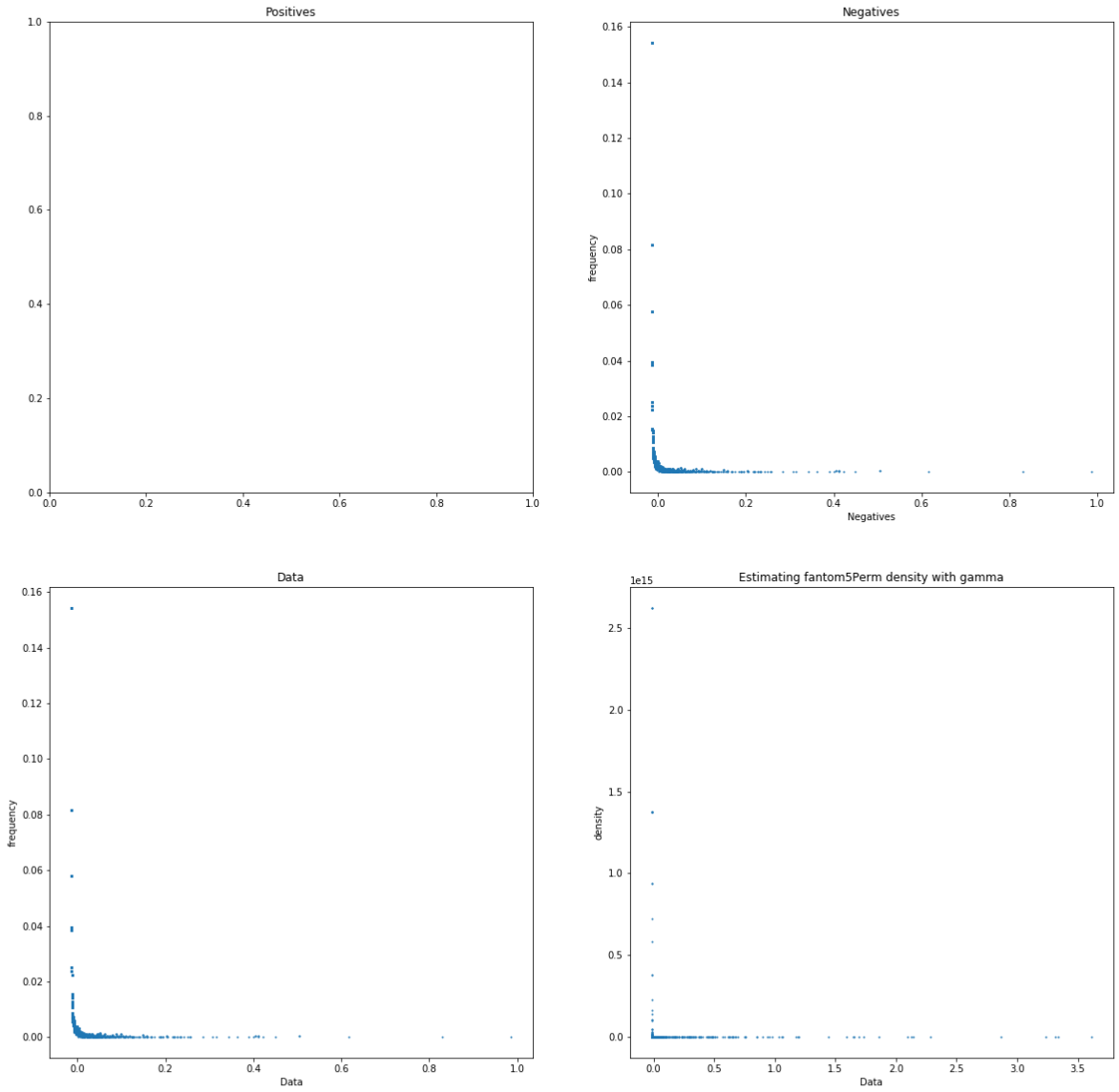


Figura 2.32: Sampling distribution of metric fantom5Perm

2.17.2 Metric values

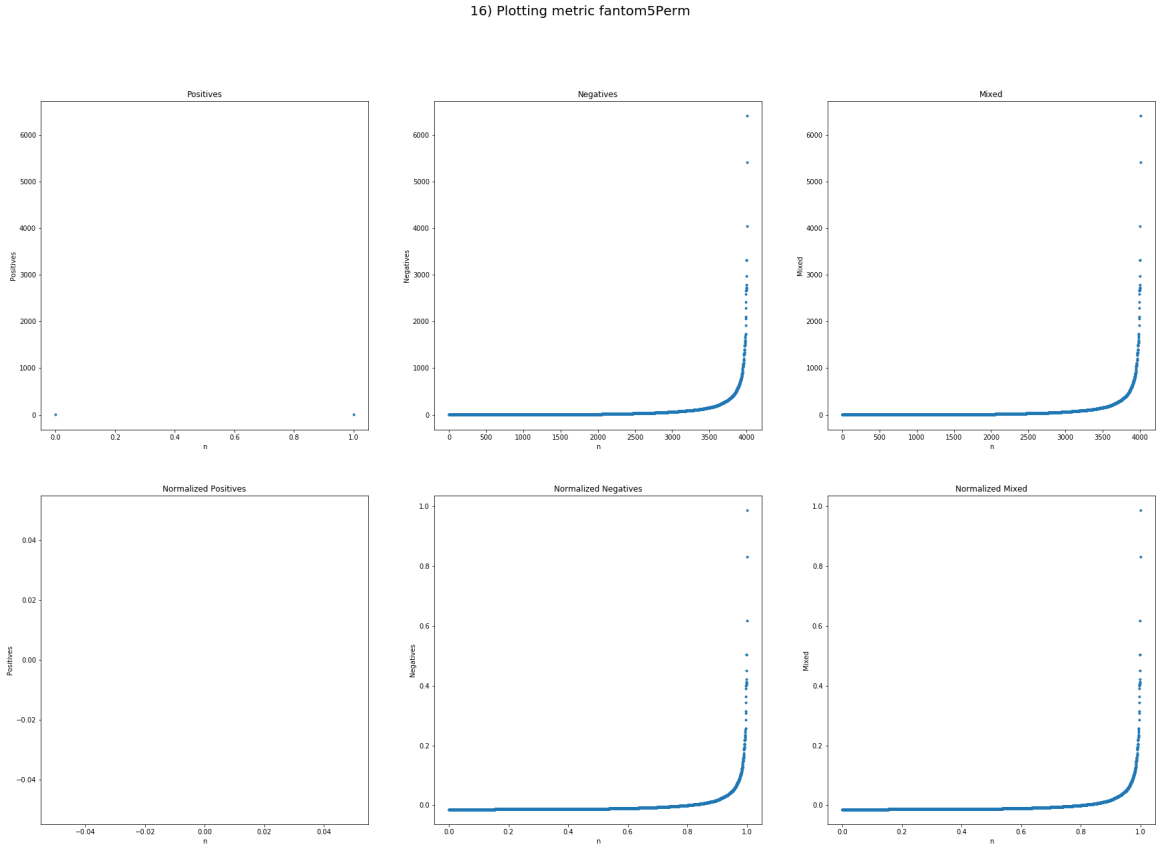


Figura 2.33: Values of metric fantom5Perm

2.18 fantom5Robust

2.18.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

17) Plotting metric fantom5Robust

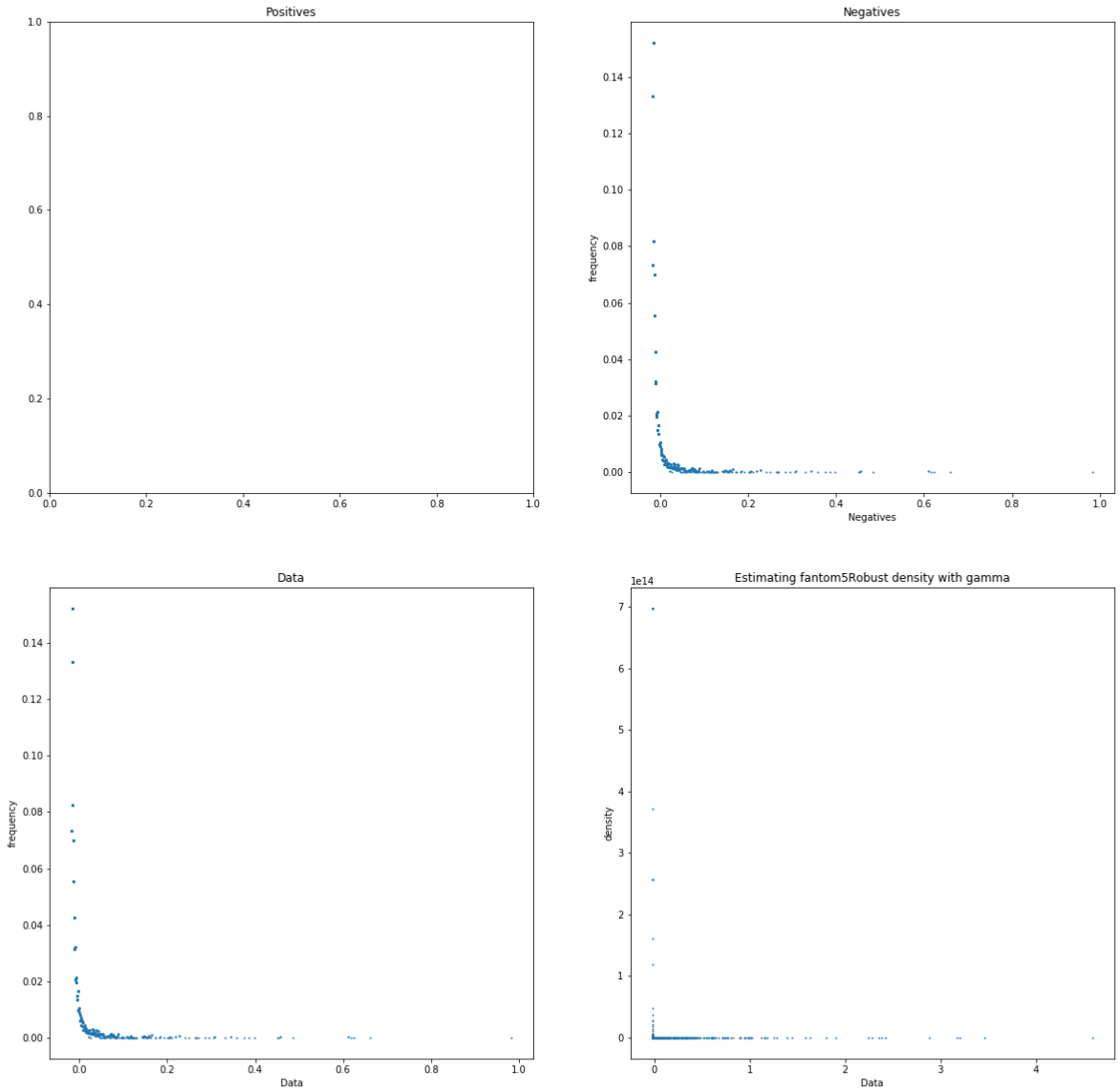


Figura 2.34: Sampling distribution of metric fantom5Robust

2.18.2 Metric values

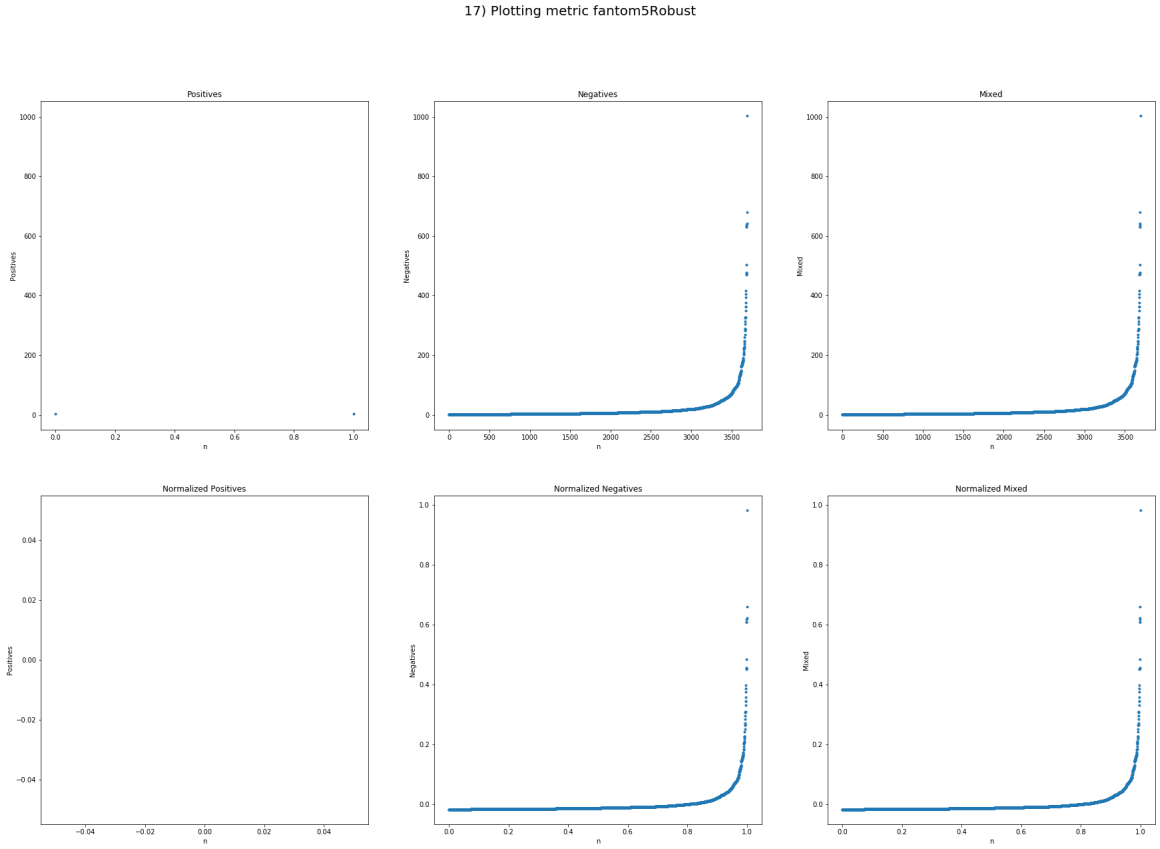


Figura 2.35: Values of metric fantom5Robust

2.19 fracRareCommon

2.19.1 Metric sample distribution

The data points seem to follow an **Beta** distribution.

18) Plotting metric fracRareCommon

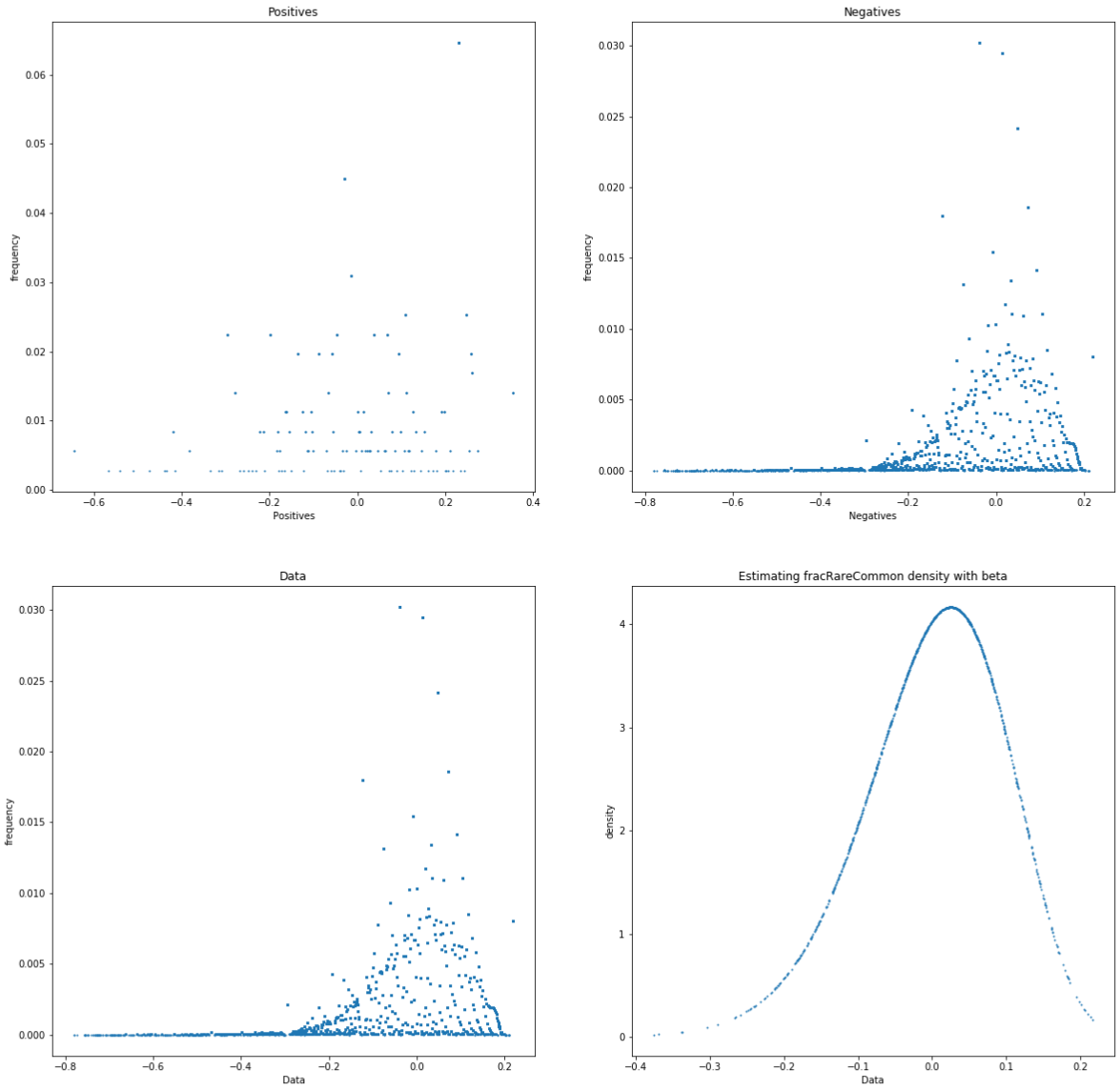


Figura 2.36: Sampling distribution of metric fracRareCommon

2.19.2 Metric values

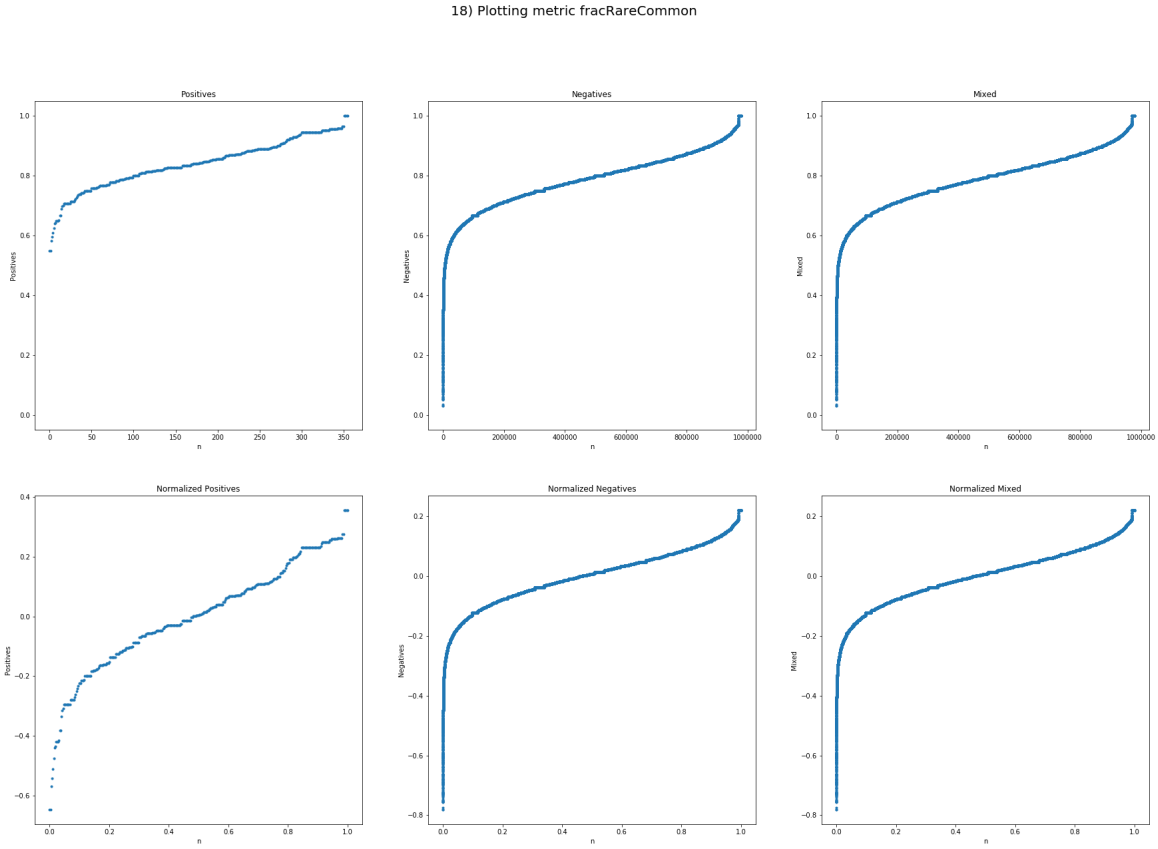


Figura 2.37: Values of metric fracRareCommon

2.20 mamPhastCons46way

2.20.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

19) Plotting metric mamPhastCons46way

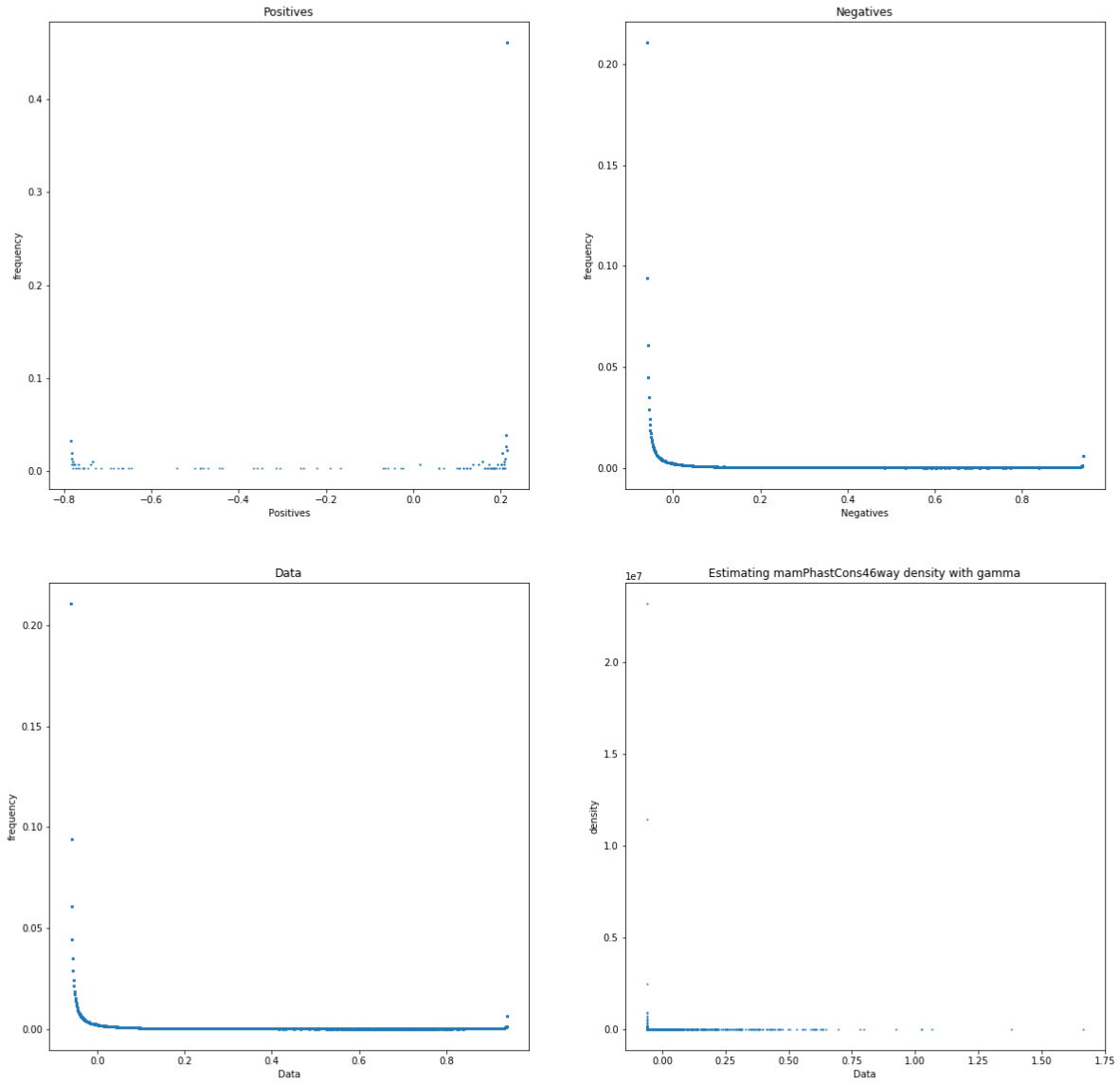


Figura 2.38: Sampling distribution of metric mamPhastCons46way

2.20.2 Metric values

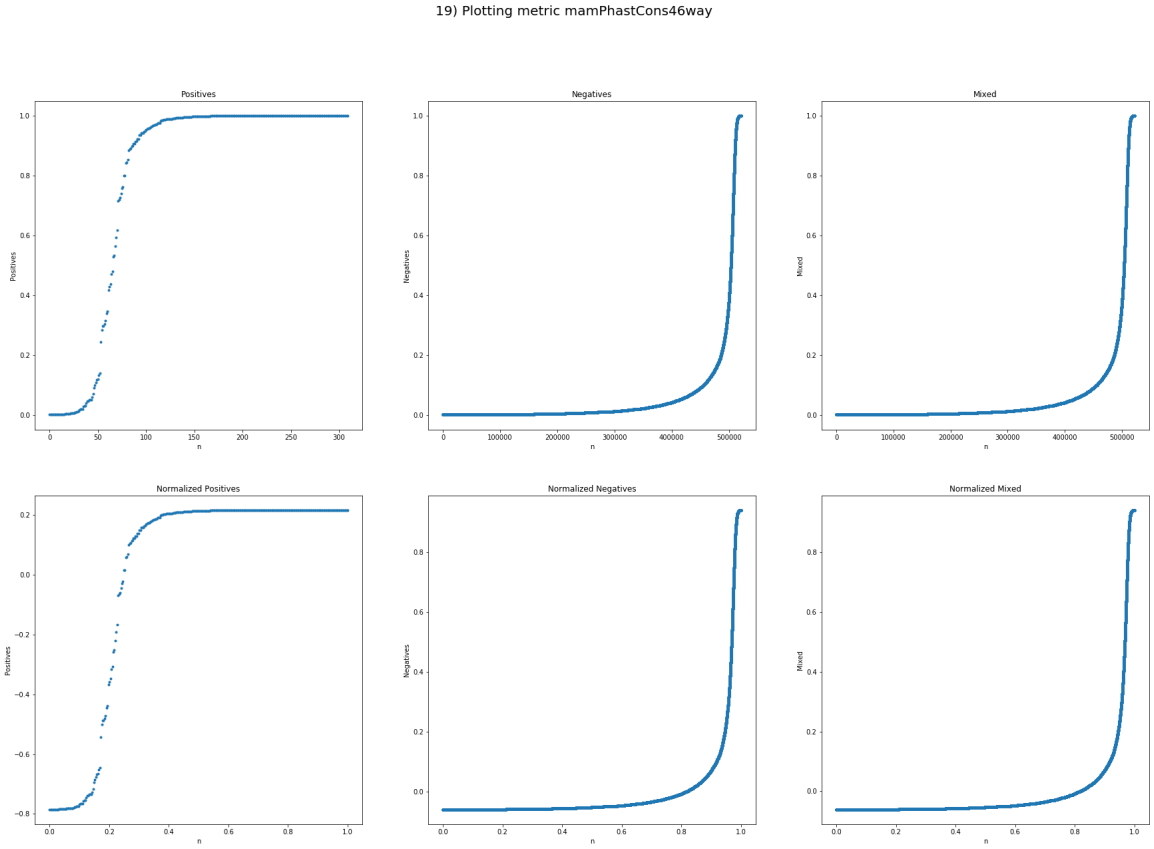


Figura 2.39: Values of metric mamPhastCons46way

2.21 mamPhyloP46way

2.21.1 Metric sample distribution

The data points seem to follow a **Gaussian** distribution.

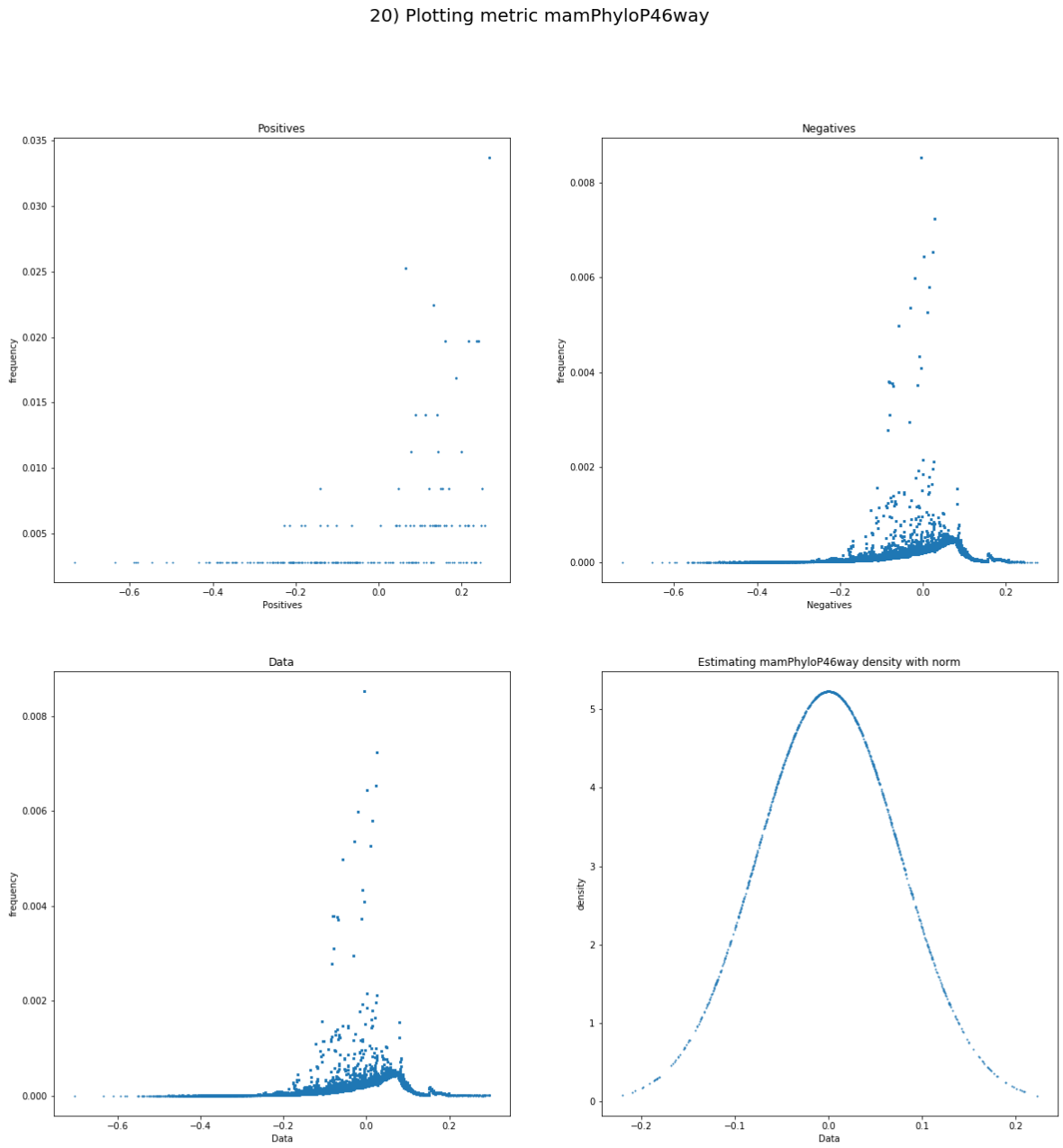


Figura 2.40: Sampling distribution of metric mamPhyloP46way

2.21.2 Metric values

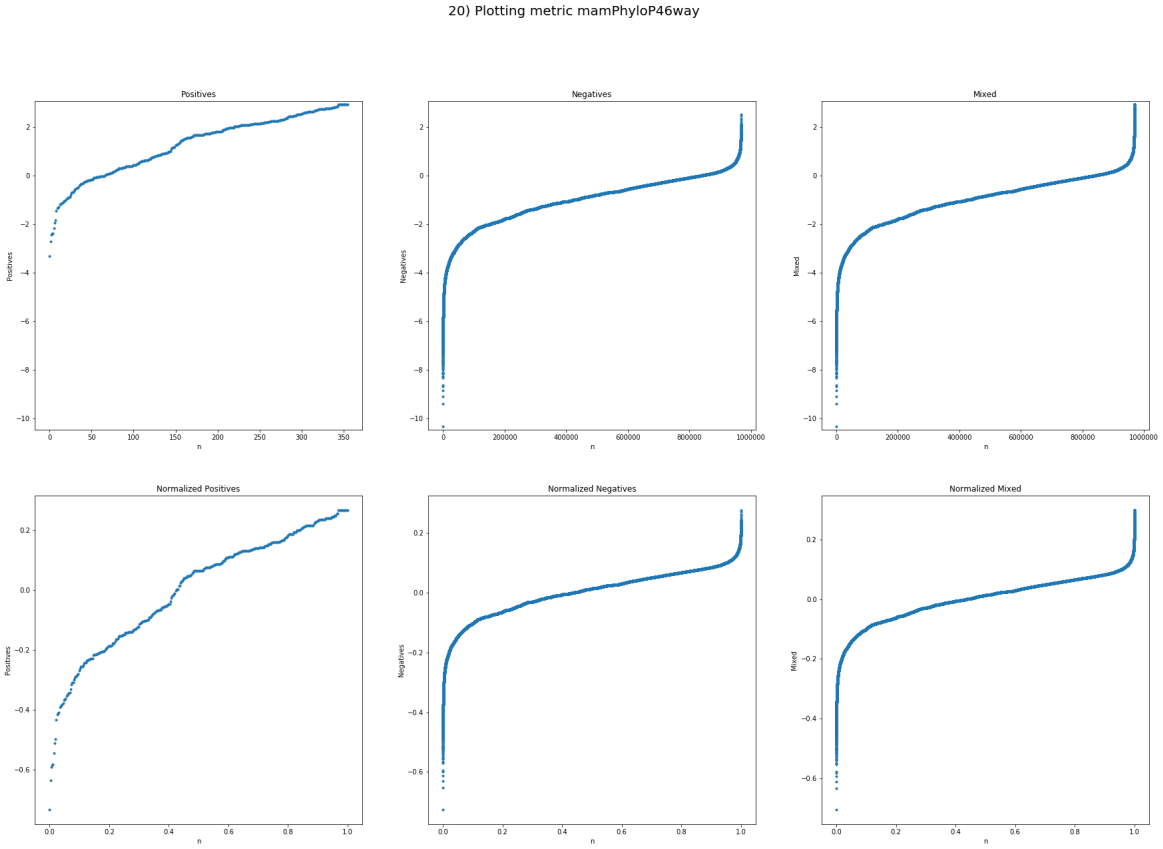


Figura 2.41: Values of metric mamPhyloP46way

2.22 numTFBSConserved

2.22.1 Metric sample distribution

The data points seem to follow a **exponential** distribution.

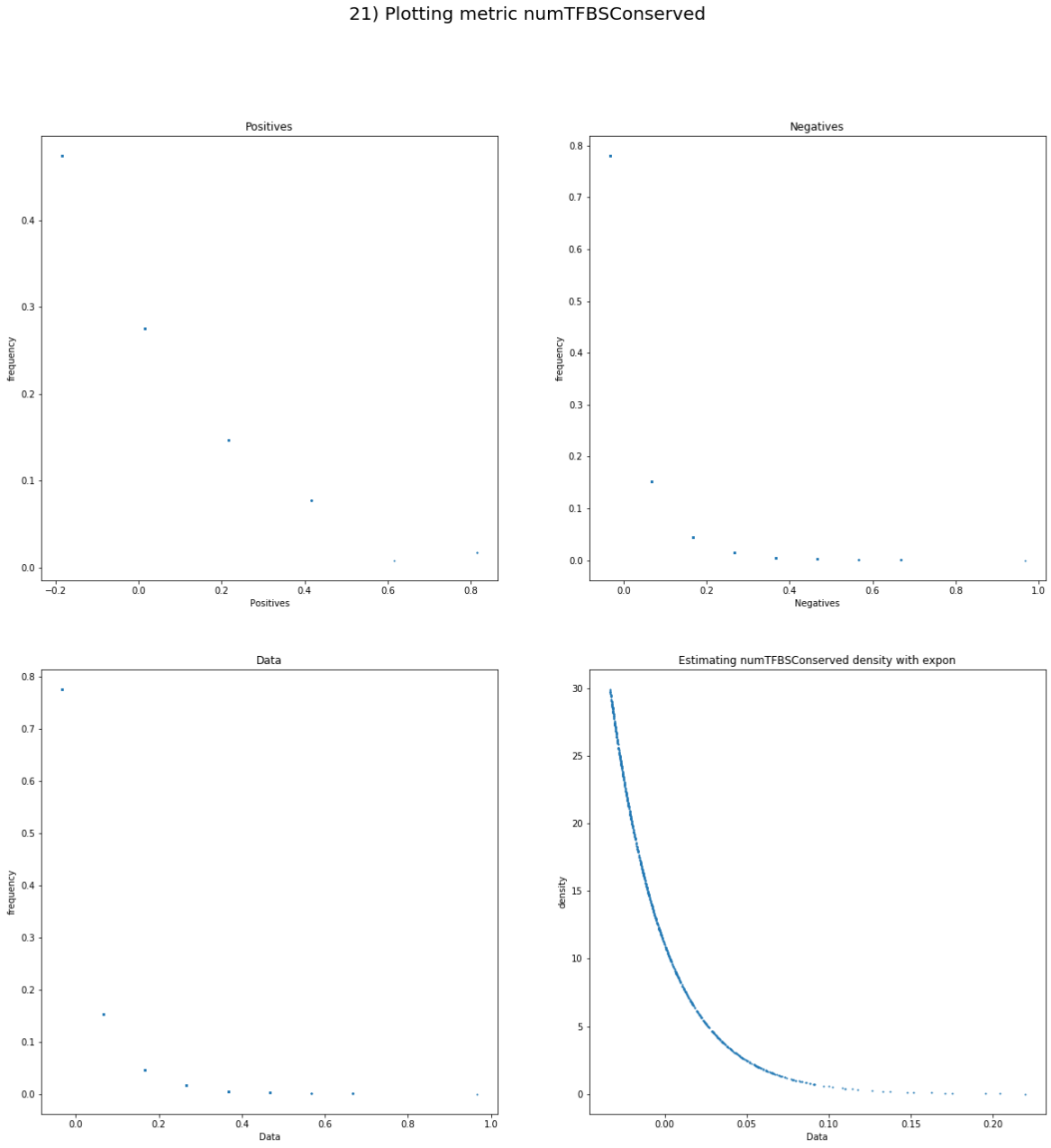


Figura 2.42: Sampling distribution of metric numTFBSConserved

2.22.2 Metric values

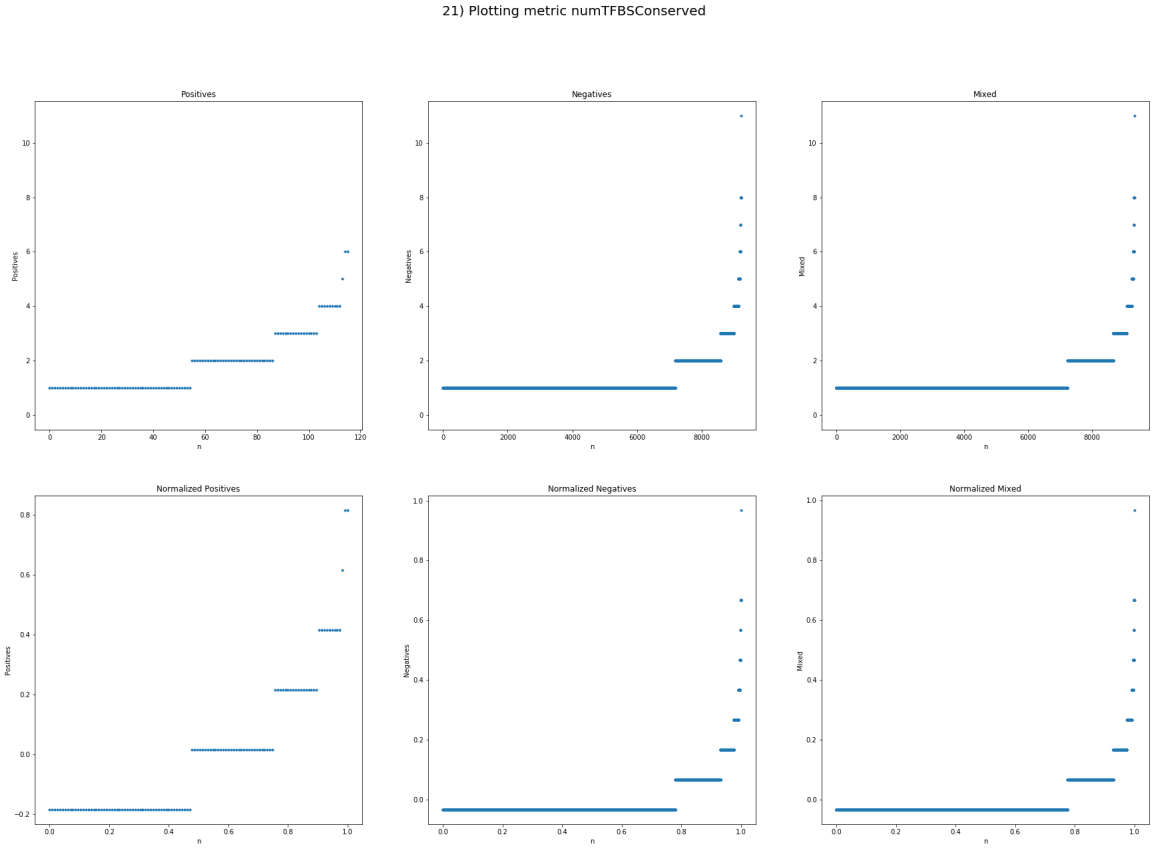


Figura 2.43: Values of metric numTFBSConserved

2.23 priPhastCons46way

2.23.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

22) Plotting metric priPhastCons46way

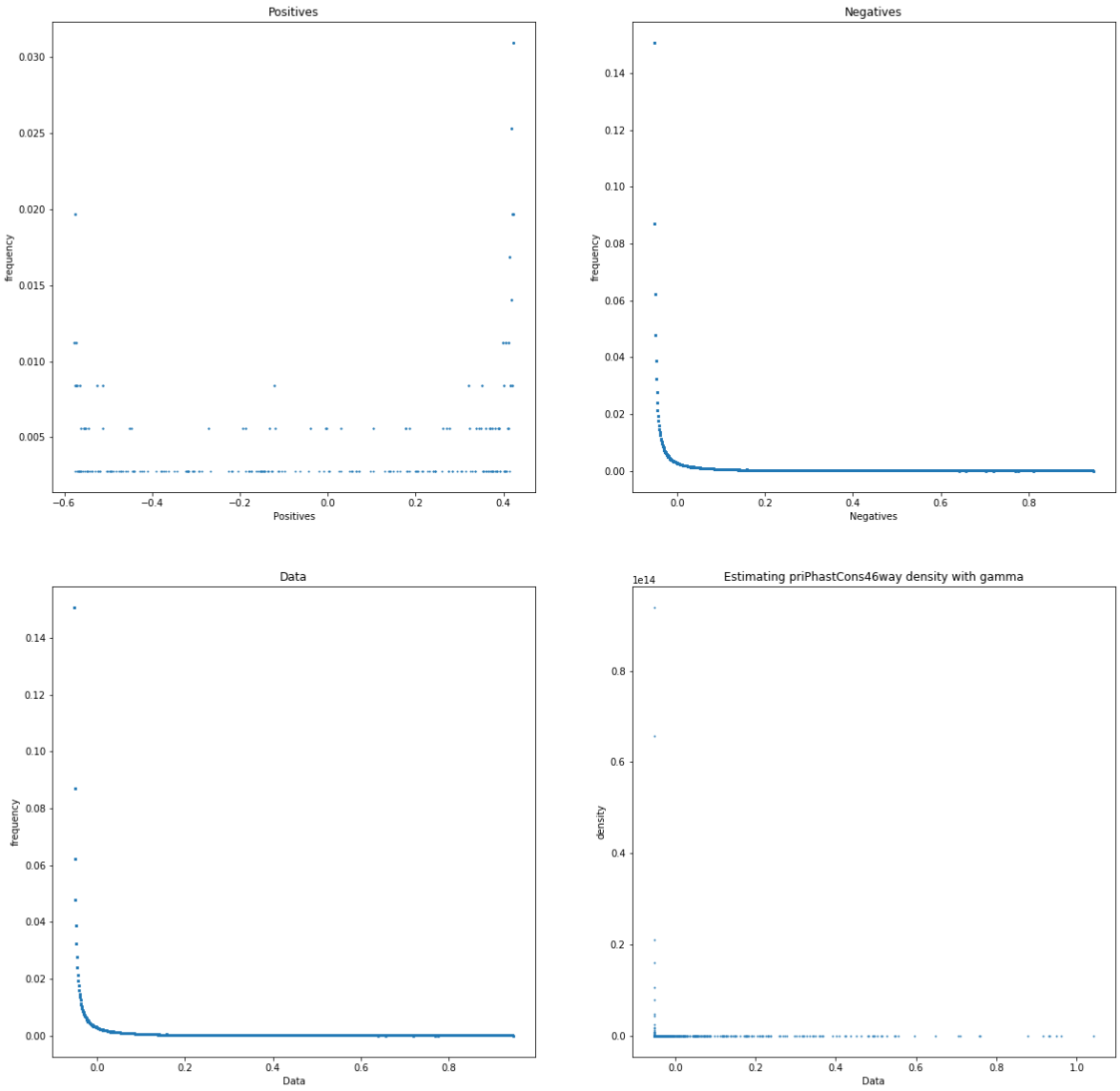


Figura 2.44: Sampling distribution of metric priPhastCons46way

2.23.2 Metric values

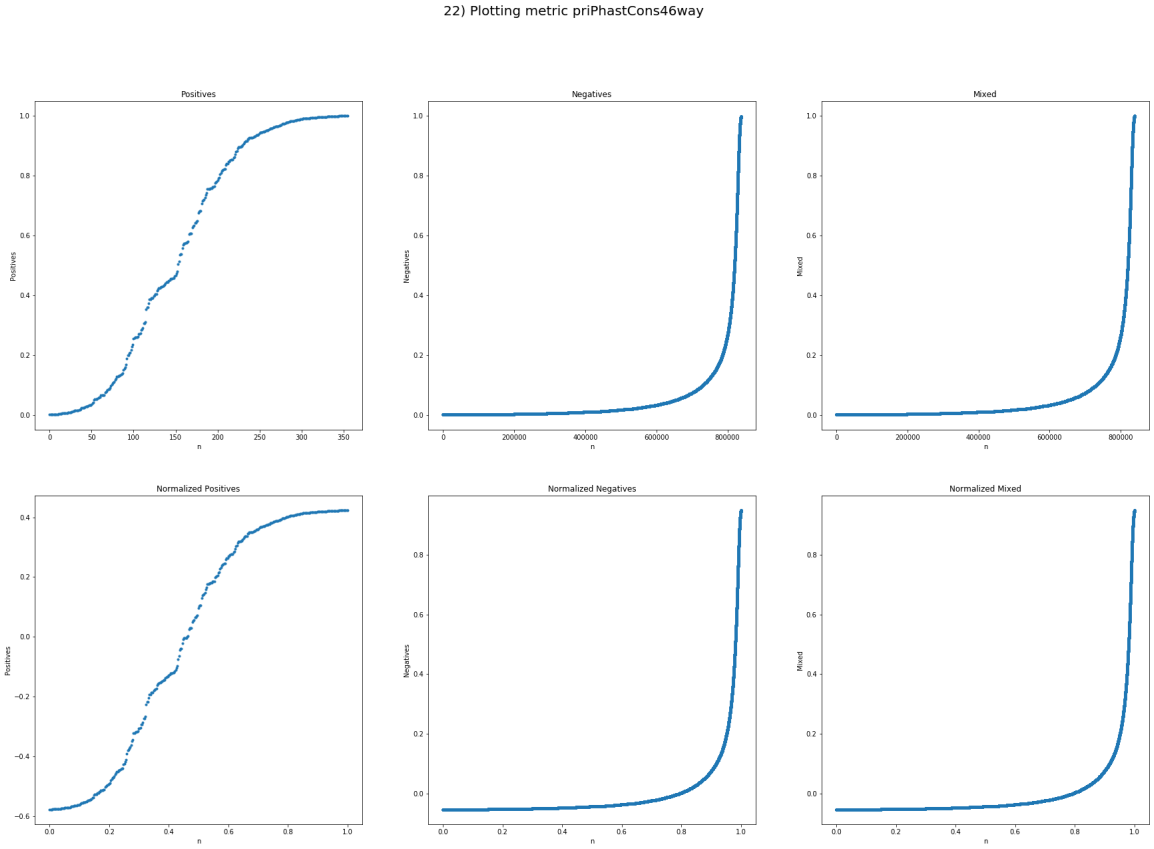


Figura 2.45: Values of metric priPhastCons46way

2.24 priPhyloP46way

2.24.1 Metric sample distribution

The data points seem to follow an **Beta** distribution.

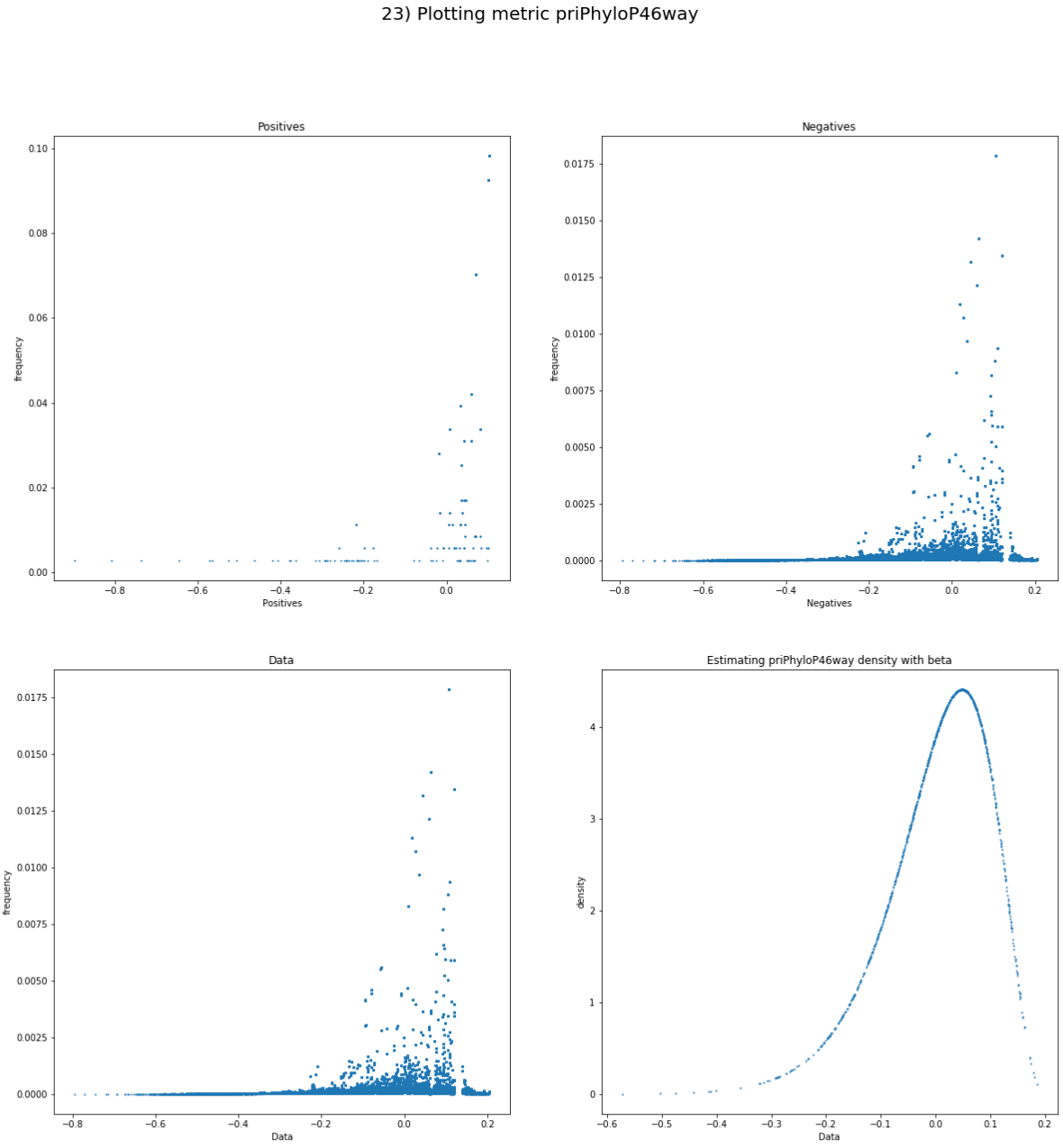


Figura 2.46: Sampling distribution of metric priPhyloP46way

2.24.2 Metric values

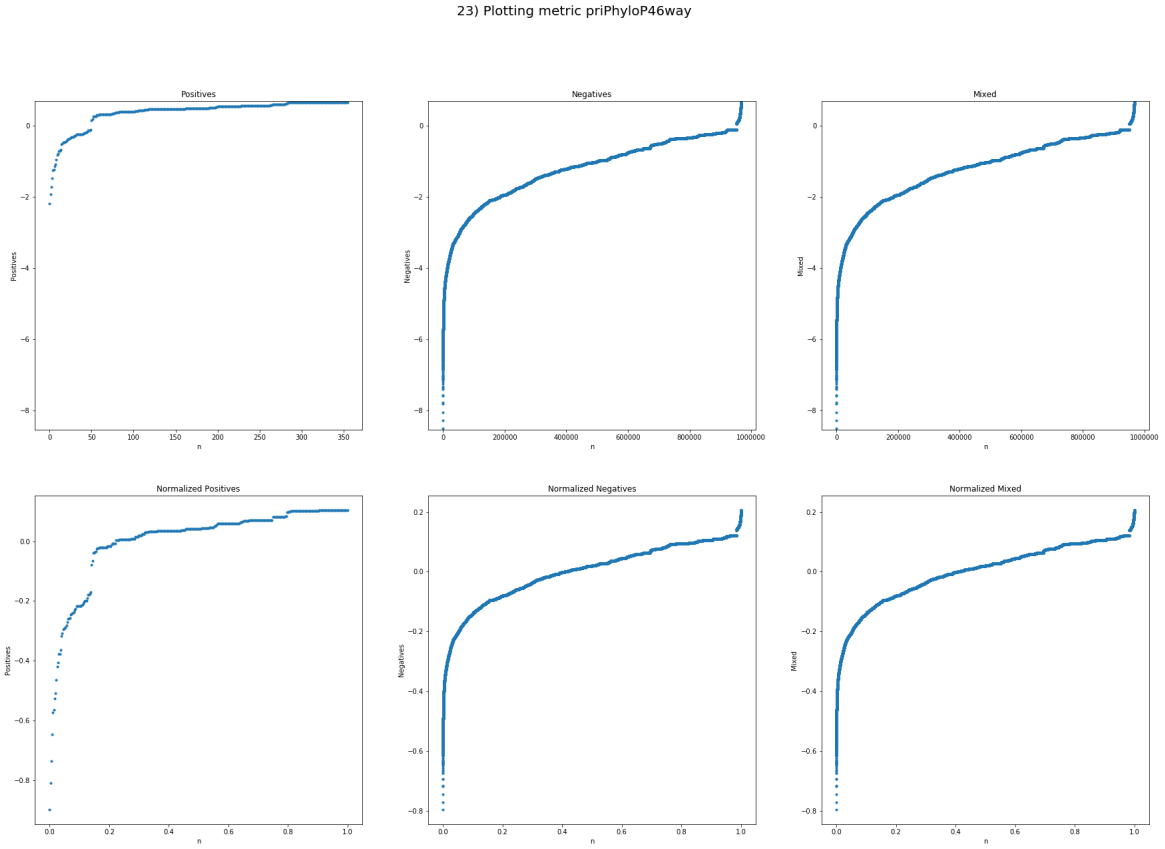


Figura 2.47: Values of metric priPhyloP46way

2.25 rareVar

2.25.1 Metric sample distribution

The data points seem to follow an **Beta** distribution.

24) Plotting metric rareVar

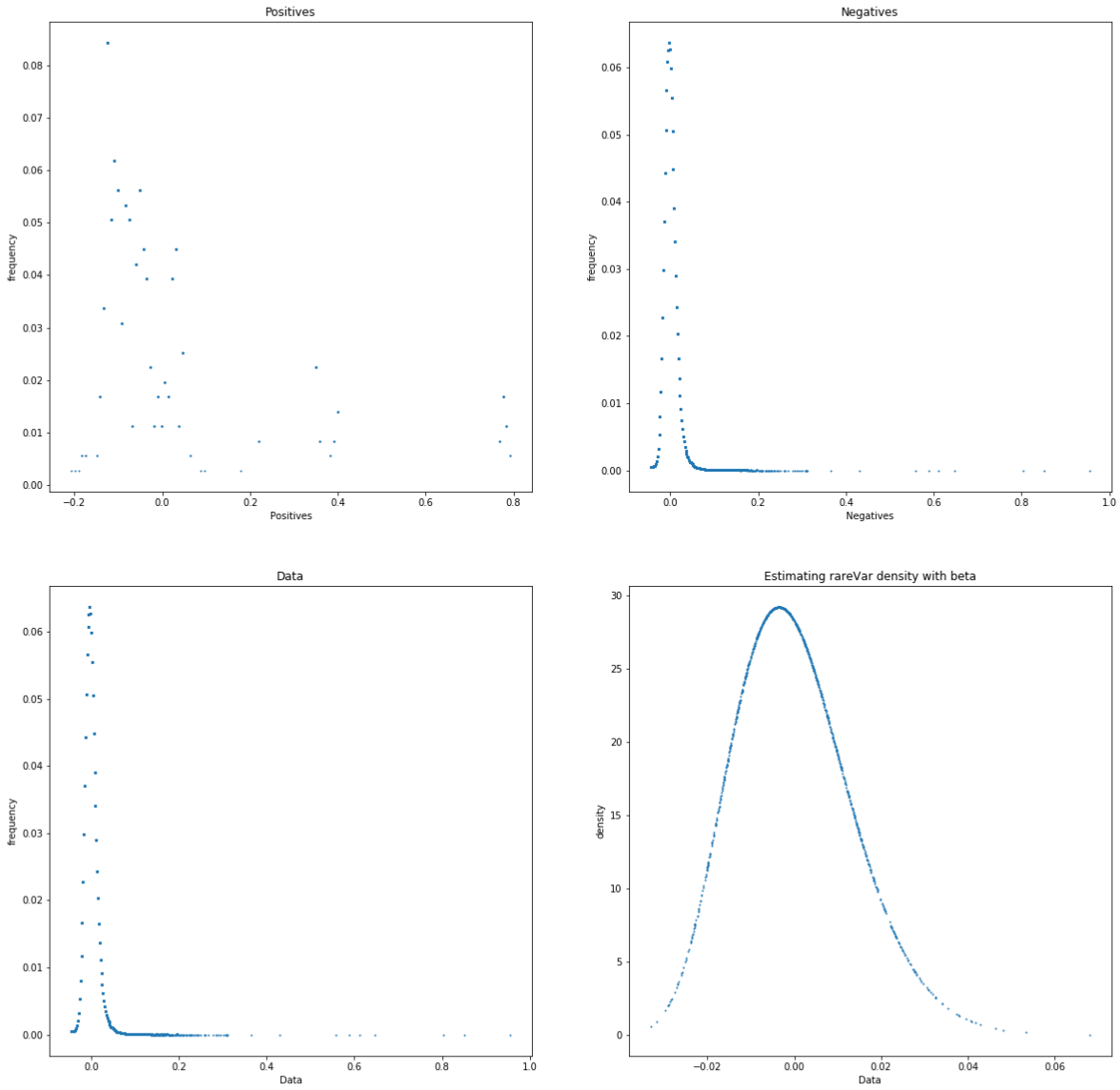


Figura 2.48: Sampling distribution of metric rareVar

2.25.2 Metric values

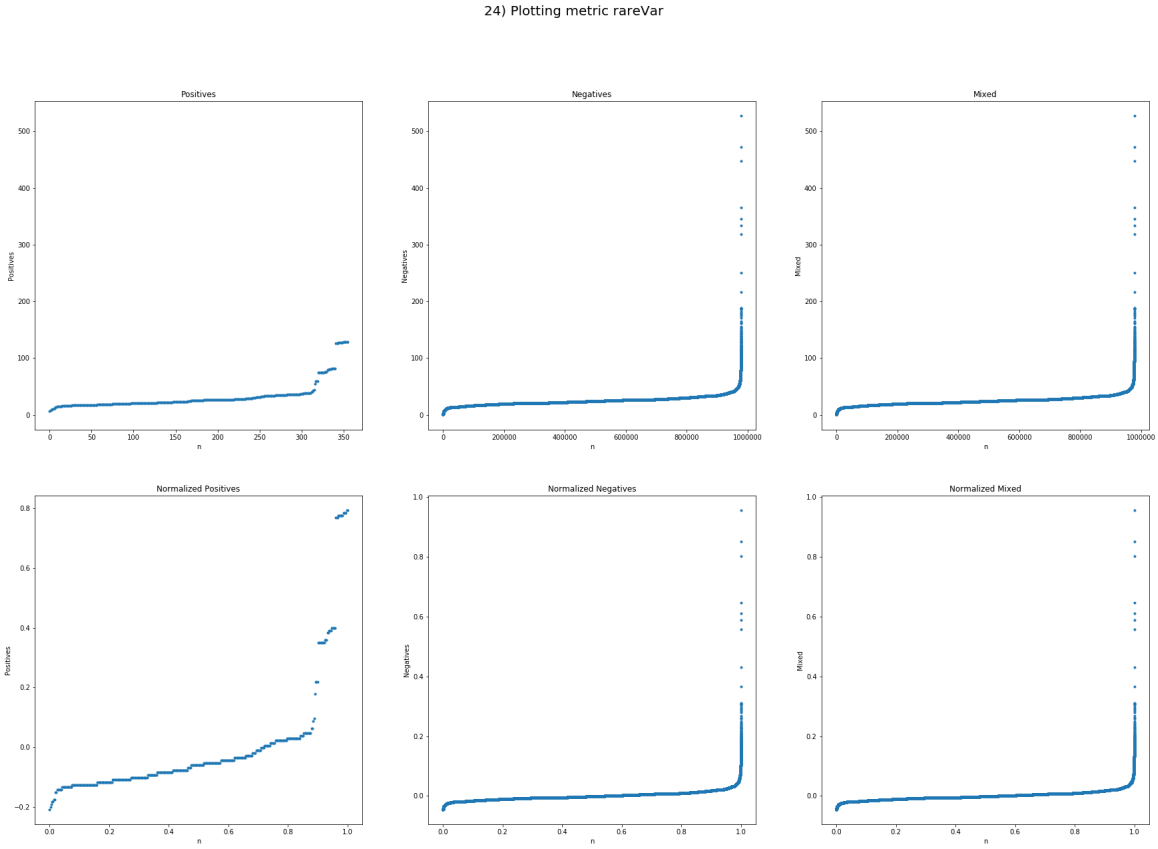


Figura 2.49: Values of metric rareVar

2.26 verPhastCons46way

2.26.1 Metric sample distribution

The data points seem to follow a **Gamma** distribution.

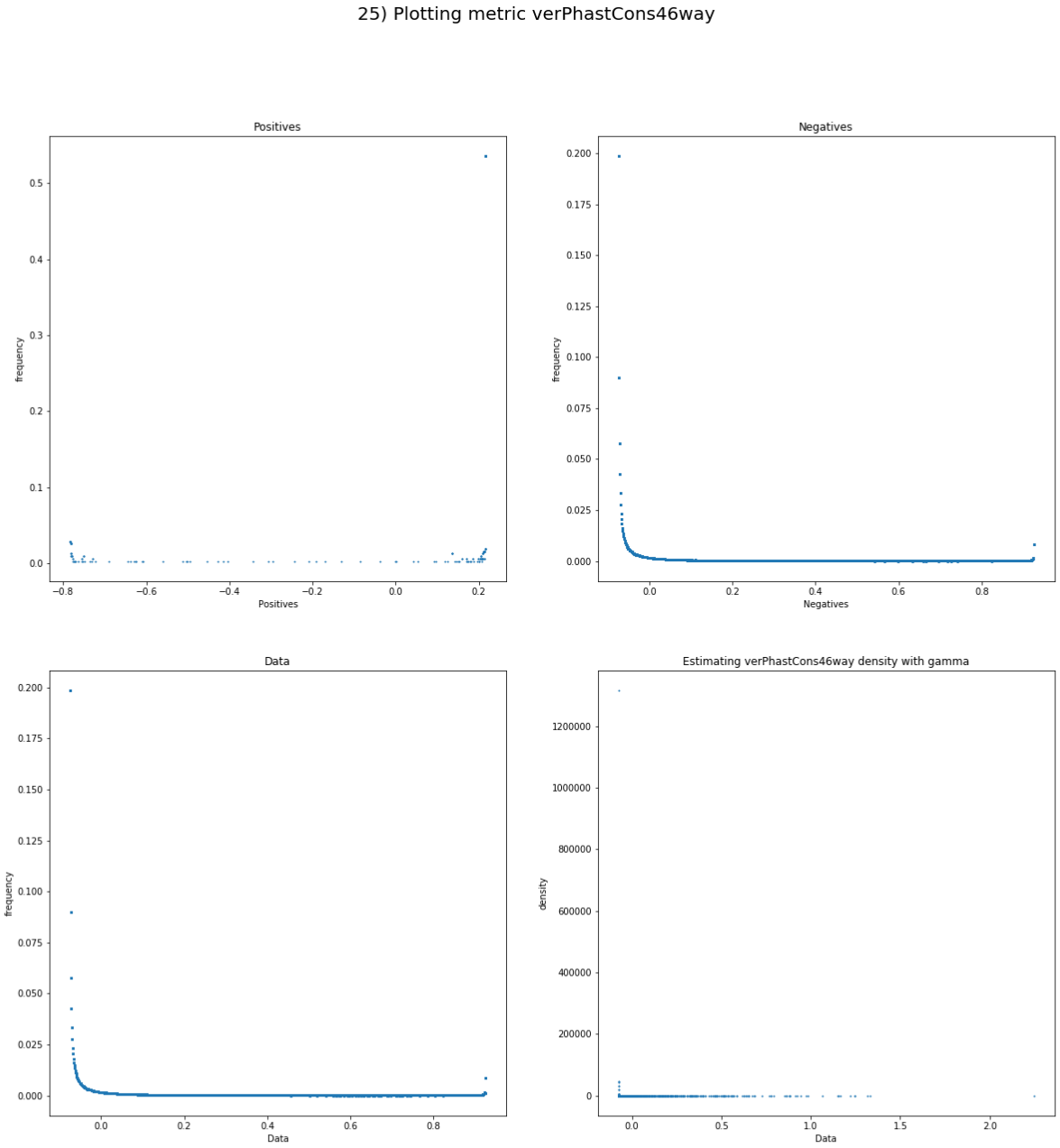


Figura 2.50: Sampling distribution of metric verPhastCons46way

2.26.2 Metric values

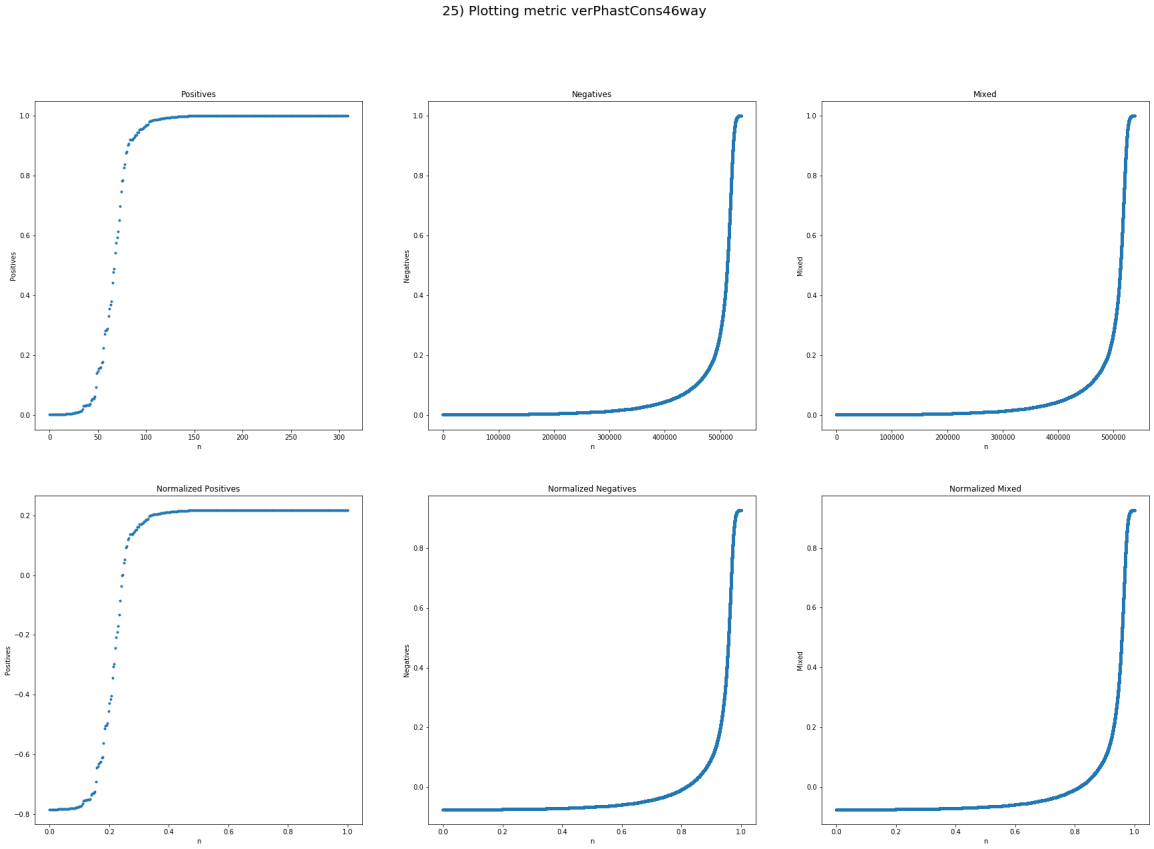


Figura 2.51: Values of metric verPhastCons46way

2.27 verPhyloP46way

2.27.1 Metric sample distribution

The data points seem to follow a **Gaussian** distribution.

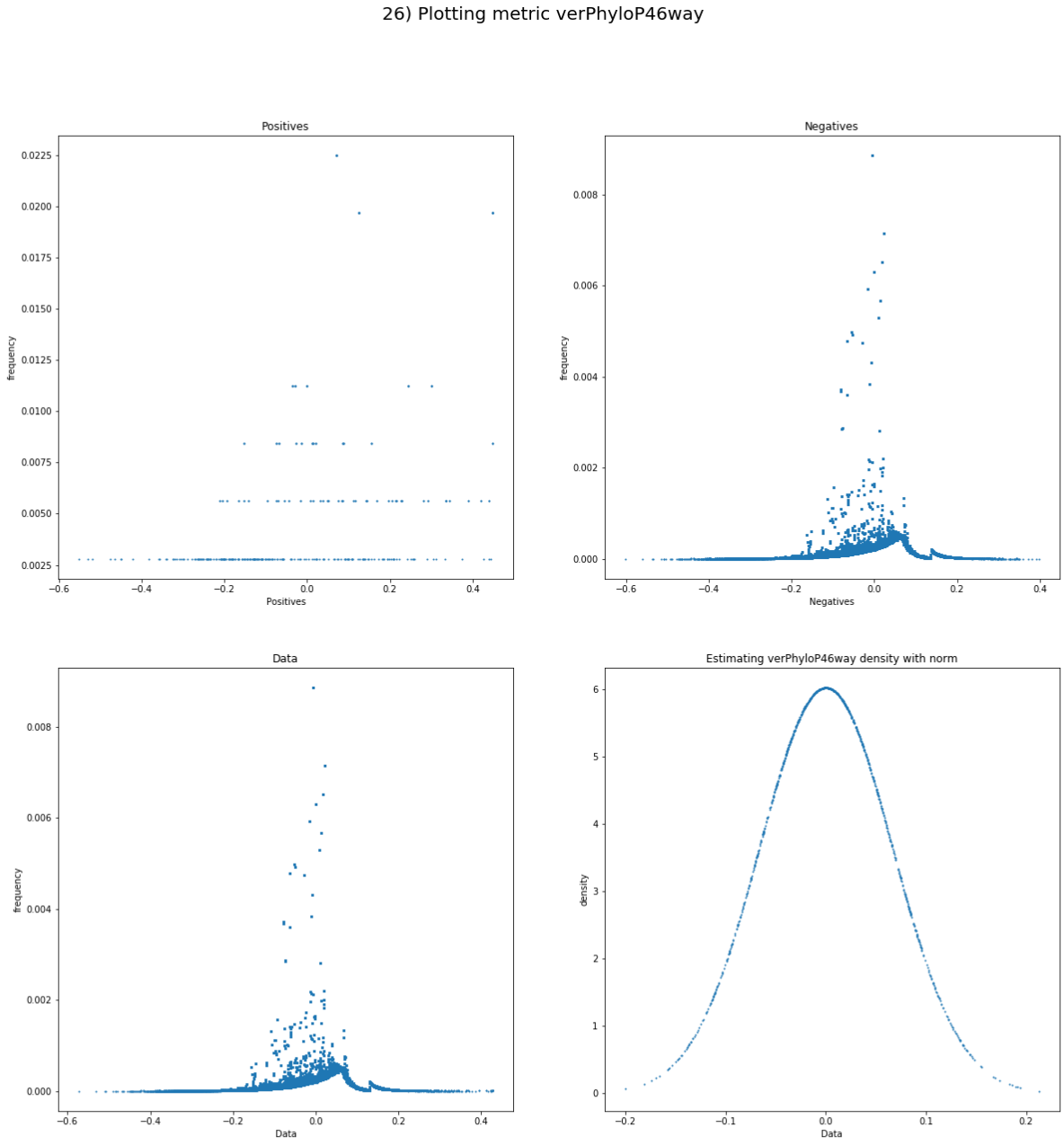


Figura 2.52: Sampling distribution of metric verPhyloP46way

2.27.2 Metric values

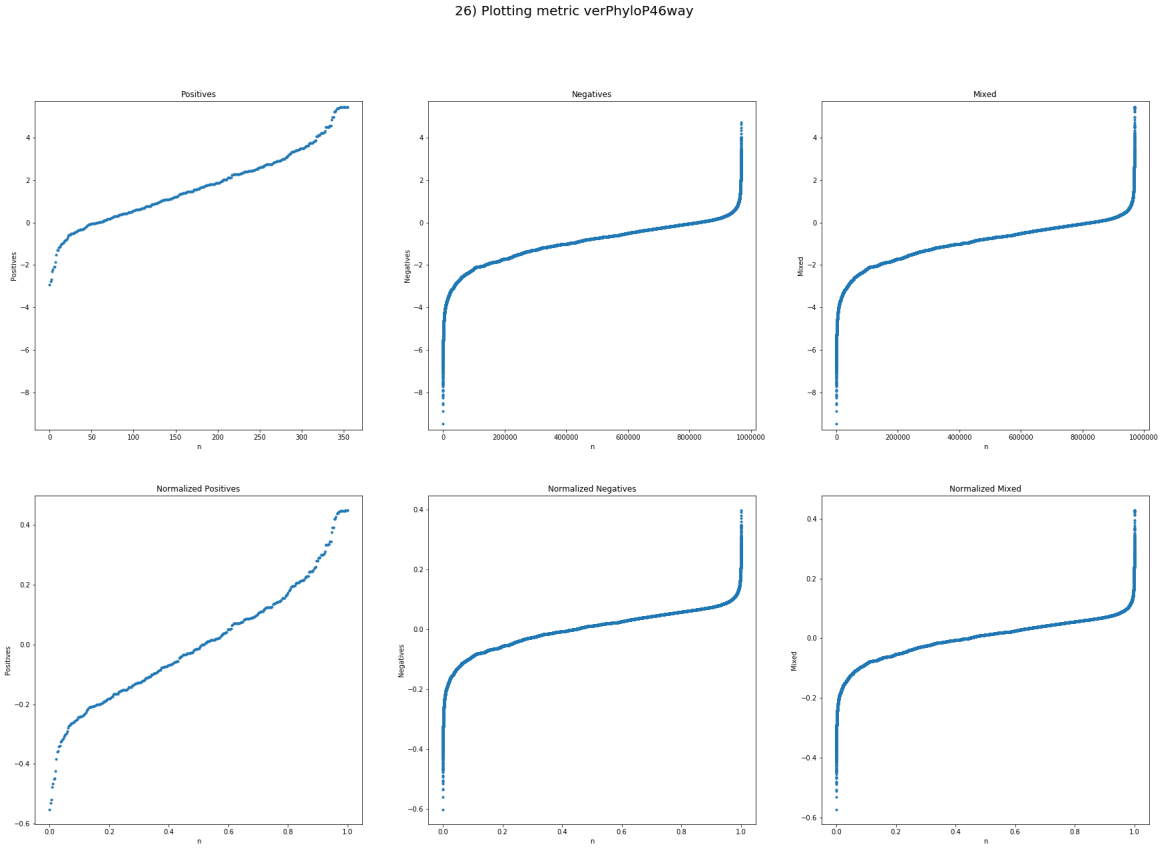


Figura 2.53: Values of metric verPhyloP46way

3

Metric distribution summary

The metrics seem to follow these sample distributions:

Metric	**Distribution**
CpGobsExp	Beta
CpGperCpG	Beta
CpGperGC	Gaussian
DGVCount	Gamma
DnaseClusteredHyp	Gamma
EncH3K27Ac	Gamma
GCContent	Gaussian
EncH3K4Me3	Gamma
ISCApath	Gamma
DnaseClusteredScore	Beta
EncH3K4Me1	Gamma
GerpRS	Gamma
GerpRSpv	Gamma
commonVar	Exponential Weibull
dbVARCount	Gamma
fantom5Perm	Gamma
fantom5Robust	Gamma
mamPhastCons46way	Gamma
priPhastCons46way	Gamma
rareVar	Beta
verPhastCons46way	Gamma
numTFBSConserved	Exponential
fracRareCommon	Beta
priPhyloP46way	Beta
verPhyloP46way	Gaussian
mamPhyloP46way	Gaussian

Tabella 3.1: Metrics and their distribution

4

Data correlation

We now proceed to try and identify eventual data correlations.

4.1 Scatter plot

A scatter plot with higher resolution is available in the project repository.

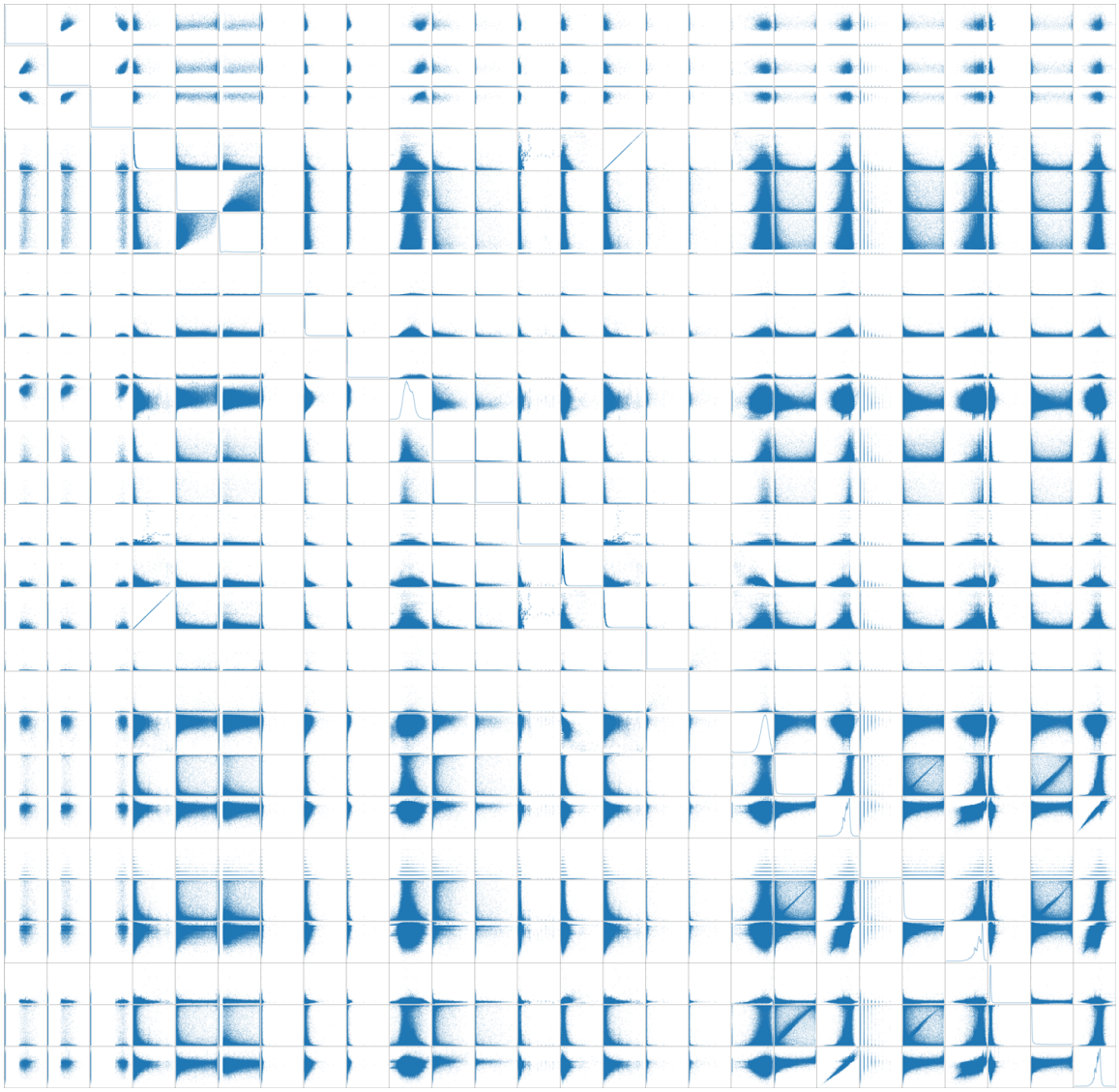


Figura 4.1: Scatter plot

4.2 Correlation coefficient matrix

A correlation matrix with higher resolution is available in the project repository.

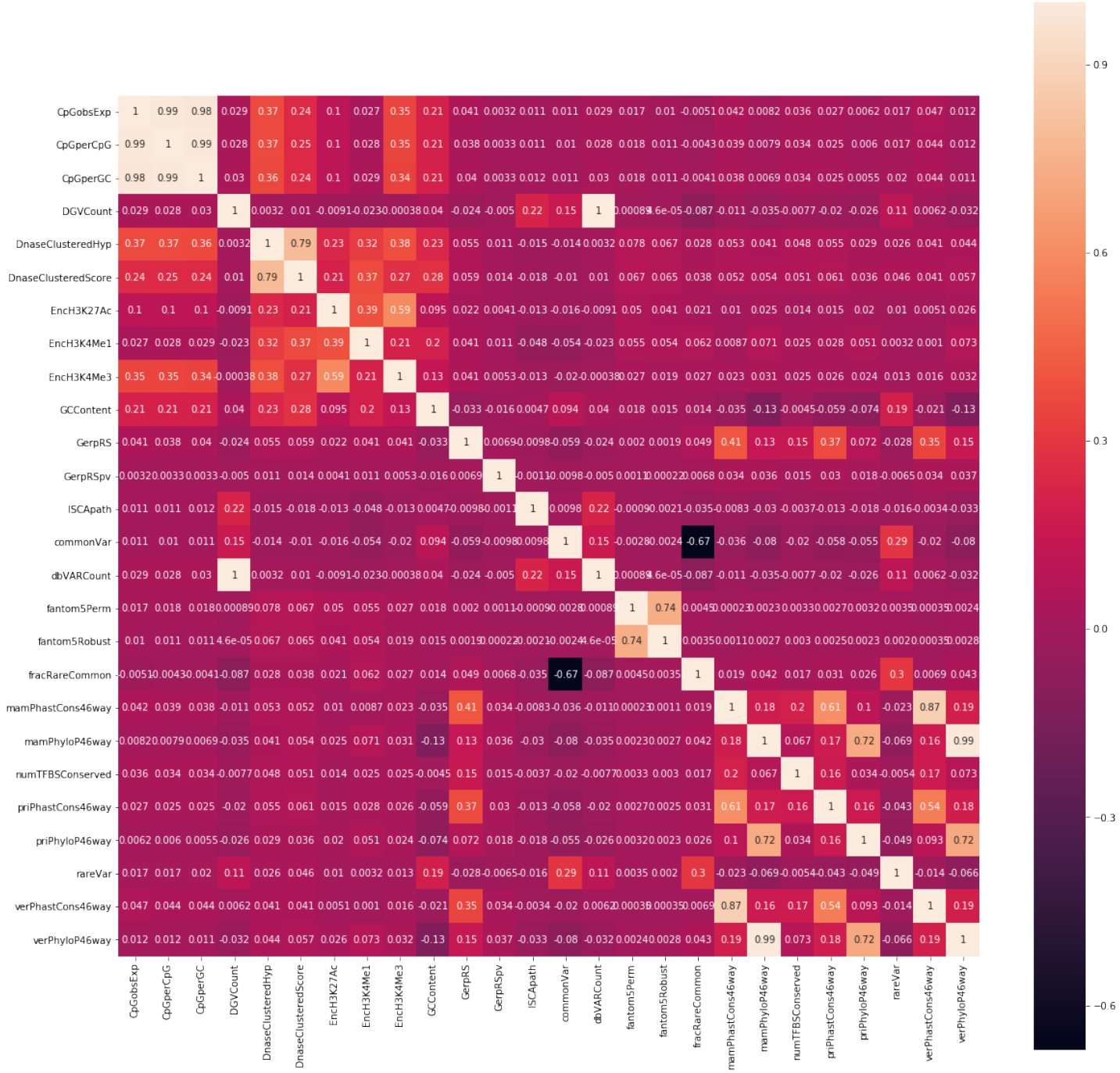


Figura 4.2: Correlation matrix

4.2.1 CpGobsExp and CpGperCpG

The two metric CpGobsExp and CpGperCpG have correlation index 0.9856203442596099.

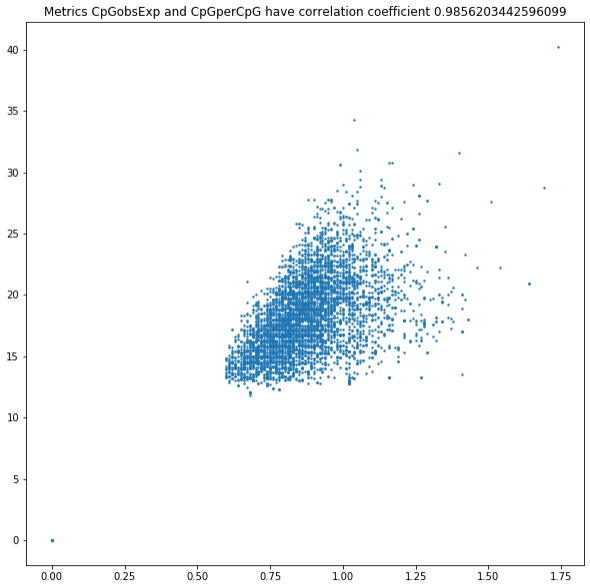


Figura 4.3: CpGobsExp and CpGperCpG

These two metrics have an extremely high correlation, so one of the two will be removed from the dataset, arbitrarily **CpGobsExp**.

4.2.2 CpGobsExp and CpGperGC

The two metric CpGobsExp and CpGperGC have correlation index 0.9785748818167331.

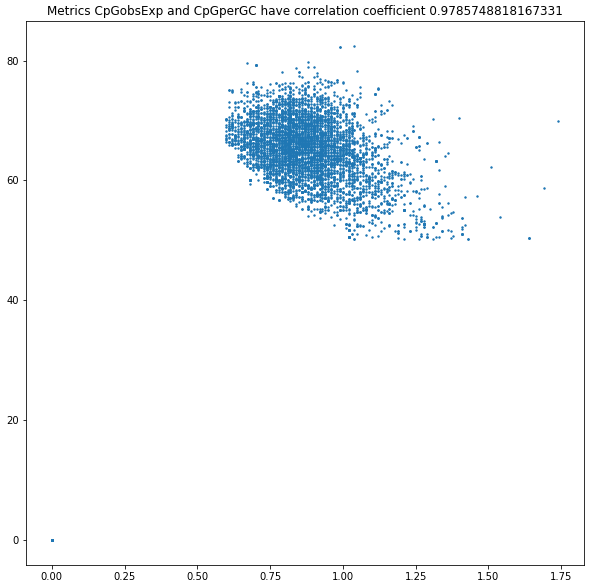


Figura 4.4: CpGobsExp and CpGperGC

These two metrics have an extremely high correlation, so one of the two will be removed from the dataset, arbitrarily **CpGobsExp**.

4.2.3 CpGperCpG and CpGperGC

The two metric CpGperCpG and CpGperGC have correlation index 0.9897887253514923.

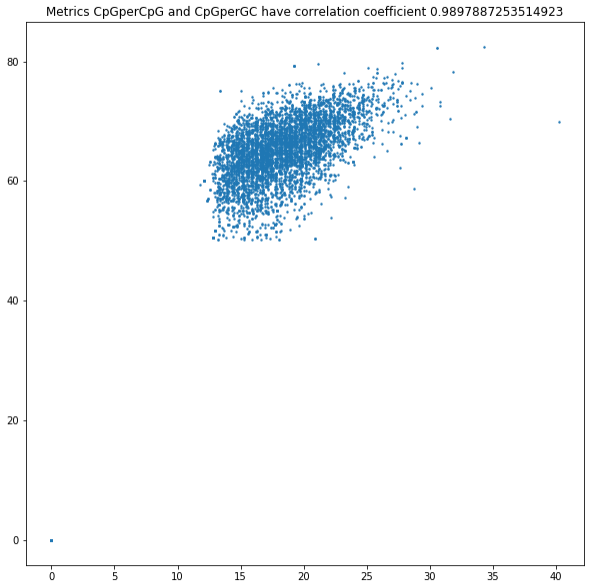


Figura 4.5: CpGperCpG and CpGperGC

These two metrics have an extremely high correlation, so one of the two will be removed from the dataset, arbitrarily **CpGperCpG**.

4.2.4 dbVARCount and DGVCount

The two metric dbVARCount and DGVCount have correlation index 1.0.

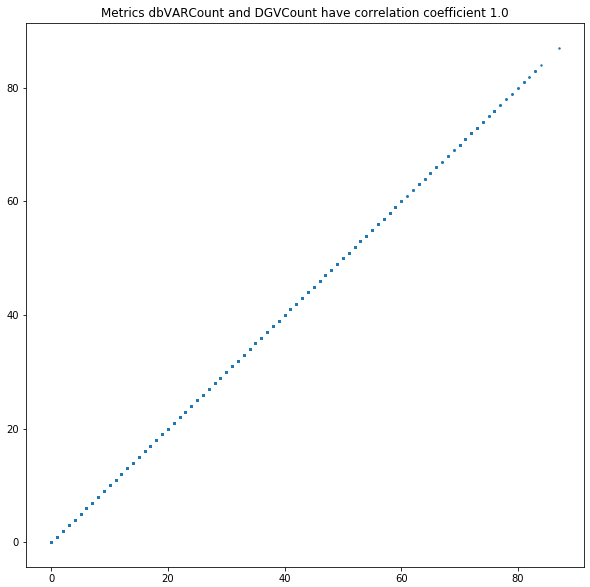


Figura 4.6: dbVARCount and DGVCount

These two metrics have an extremely high correlation, so one of the two will be removed from the dataset, arbitrarily **dbVARCount**.

4.2.5 mamPhyloP46way and verPhyloP46way

The two metric mamPhyloP46way and verPhyloP46way have correlation index 0.9902257463490804.

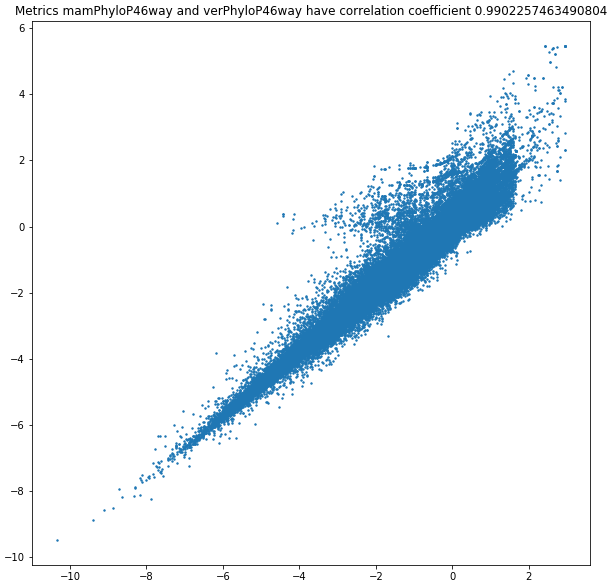


Figura 4.7: mamPhyloP46way and verPhyloP46way

These two metrics have an extremely high correlation, so one of the two will be removed from the dataset, arbitrarily **mamPhyloP46way**.

4.2.6 DnaseClusteredHyp and DnaseClusteredScore

The two metric DnaseClusteredHyp and DnaseClusteredScore have correlation index 0.7863337778663062.

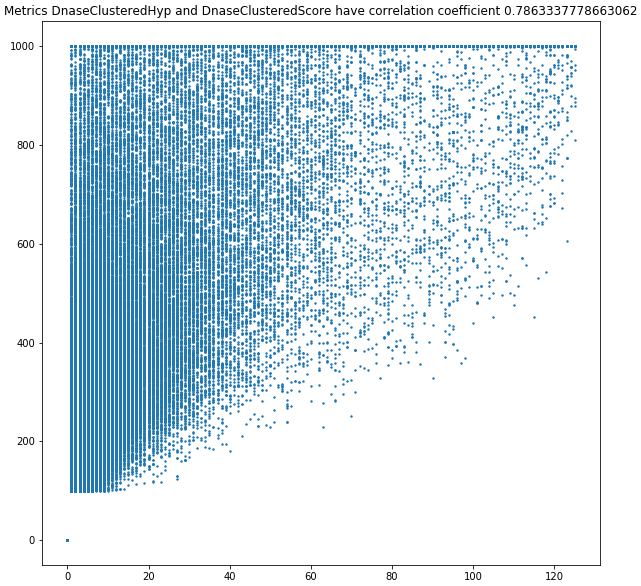


Figura 4.8: DnaseClusteredHyp and DnaseClusteredScore

The correlation value is high, but not enough to motivate actions such as the removal from the dataset.

4.2.7 mamPhastCons46way and verPhastCons46way

The two metric mamPhastCons46way and verPhastCons46way have correlation index 0.8700263713585867.

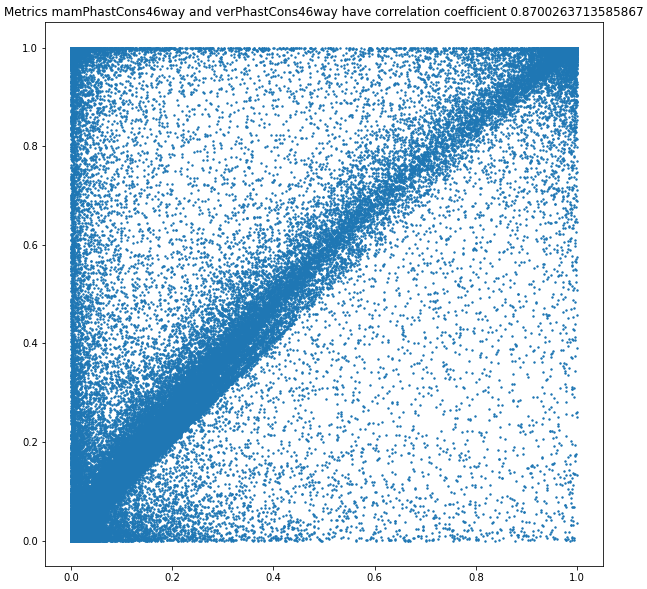


Figura 4.9: mamPhastCons46way and verPhastCons46way

The correlation value is high, but not enough to motivate actions such as the removal from the dataset.

4.3 Identified data correlations

Data correlations exist between:

First metric	Second metric	Correlation coefficient
CpGobsExp	CpGperCpG	0.9856203442596099
CpGobsExp	CpGperGC	0.9785748818167331
CpGperCpG	CpGperGC	0.9897887253514923
dbVARCount	DGVCount	1.0
mamPhyloP46way	verPhyloP46way	0.9902257463490804
DnaseClusteredHyp	DnaseClusteredScore	0.7863337778663062
mamPhastCons46way	verPhastCons46way	0.8700263713585867

4.4 Correlation table after removing highly correlated data

After having removed from the dataset metrics **CpGobsExp**, **CpGperCpG**, **dbVARCount** and **mamPhyloP46way** looks like:

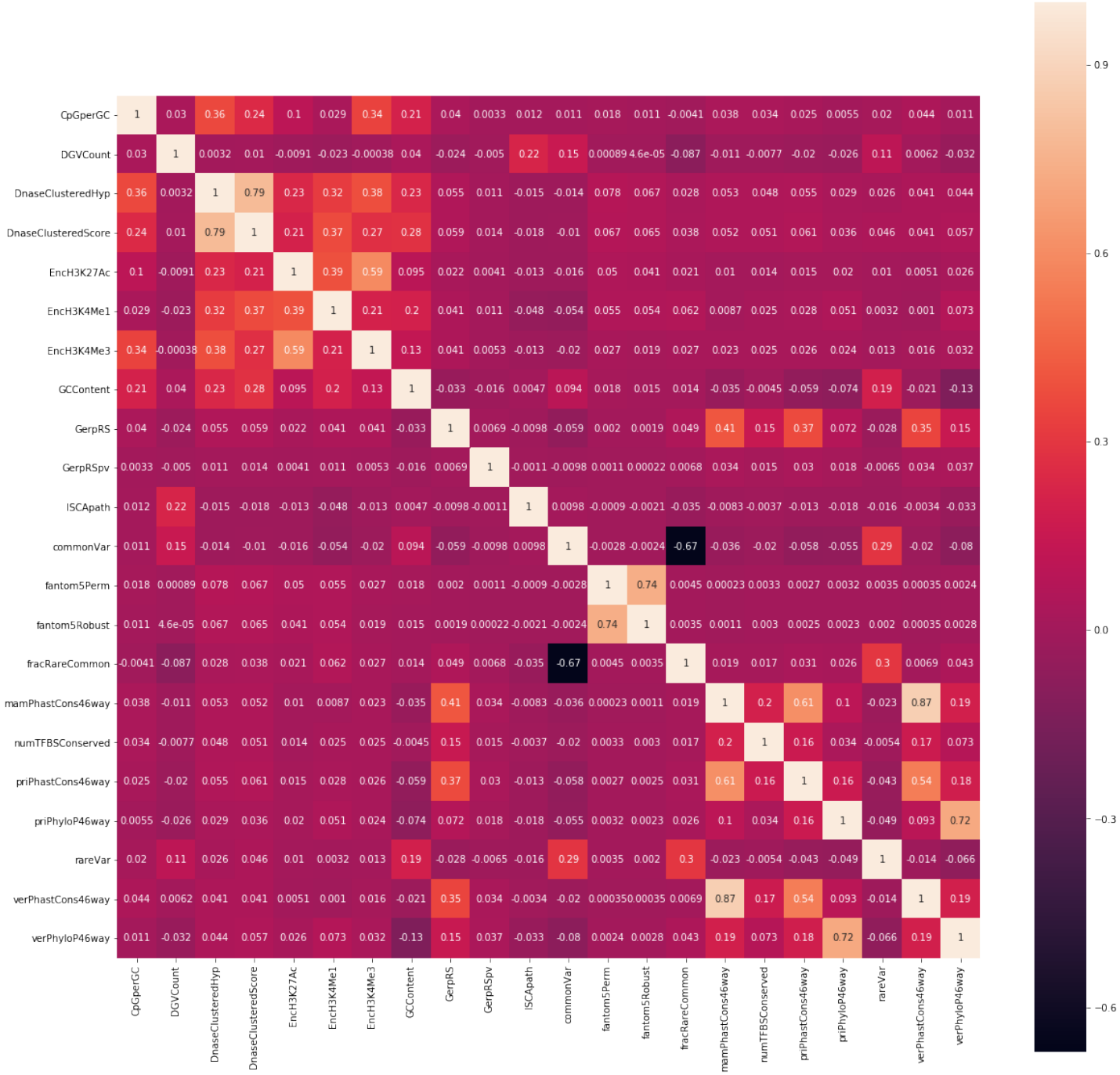


Figura 4.10: Correlation matrix

This correlation matrix with higher resolution is available in the project repository.

5 Dataset visualization

After having removed the correlated metrics we can proceed to use techniques of dimensionality reduction for visualization to see if the dataset is valid for a clustering or machine learning approach:

5.1 PCA

5.1.1 Training dataset visualization

The positive data are clustered inside the negative data: a simple clustering approach would, most probably, not be enough for separating the two classes, but easily separable for any multi-parameters ML approach like networks.

The points isolated on the right are most probably errors in the realization of the dataset.

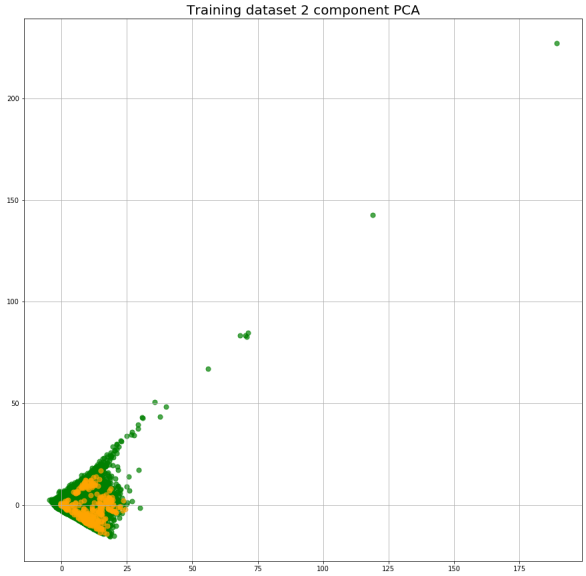


Figura 5.1: Training dataset visualization

5.1.2 Testing dataset visualization

The testing dataset is probably malformed: all the positives point are visibly clustered. A simple clustering approach such as K-Means could probably separate most of them successfully.

An ML approach with a network will probably reach sooner an high accuracy on the testing dataset than on the training dataset for this reason, being able to classify most of the positive points immediately.

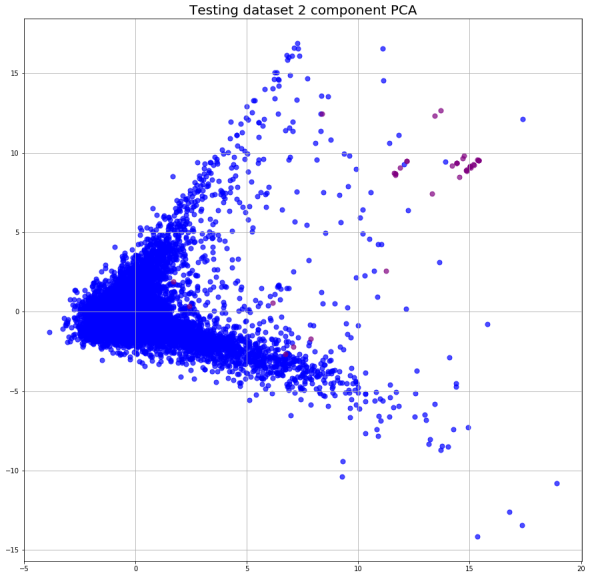


Figura 5.2: Testing dataset visualization

5.1.3 Mixed dataset visualization

The two datasets overlap correctly.

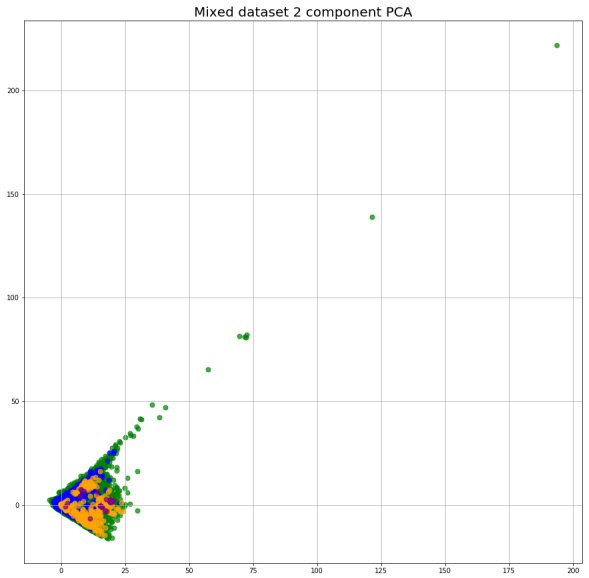


Figura 5.3: Mixed dataset visualization

5.2 TSNE

5.2.1 Training dataset visualization

Implementation exists but it is not runnable on the current machine in a decent time.

5.2.2 Testing dataset visualization

With TSNE visualization, after 1000 iterations, the training dataset positive class appears on the border right of the cluster.

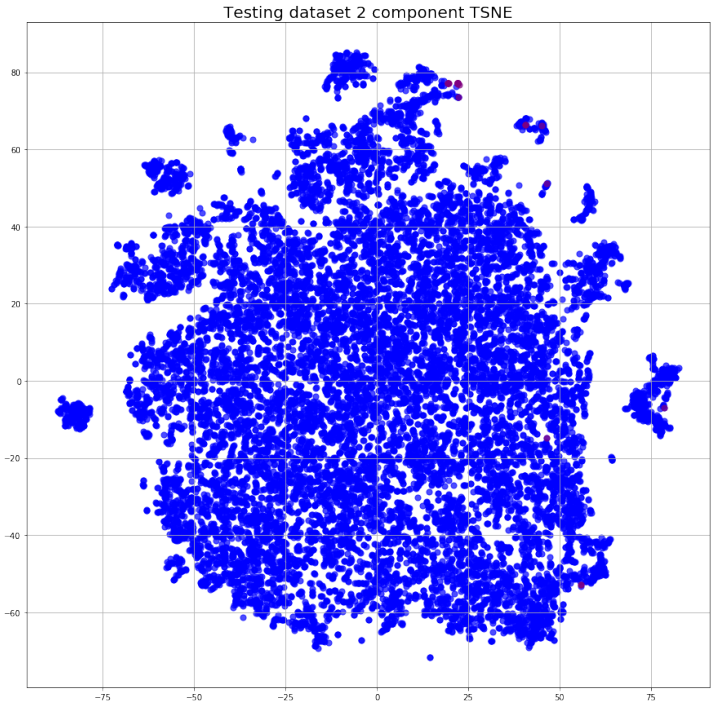


Figura 5.4: Testing dataset visualization

5.2.3 Mixed dataset visualization

Implementation exists but it is not runnable on the current machine in a decent time.

6 Dataset issues

6.1 Possible dataset errors

The PCA visualization shows points extremely out of the dataset cluster. It is possible that these points are errors.

6.2 Biased testing dataset

The testing dataset does not seem to reflect the training dataset distribution, but agglomerates the two classes in two separated clusters. It is therefore unhelpful when calculating the classification success in networks, as the training dataset result harder to classify that the testing one.

Parte II

Network implementation

7 Model architecture

Based on the informations extracted from the dataset structure, a small network should work best.

7.1 Activation function

We'll be using the **SeLU** (9) activation function for most of the layers for its property of auto-normalization.

$$\text{SeLU}(x) = \lambda \begin{cases} x & x > 0 \\ \alpha e^x - \alpha & x \leqslant 0 \end{cases}$$

Figura 7.1: SeLU

7.2 Dense layers

Dense layers will be used through all the network:

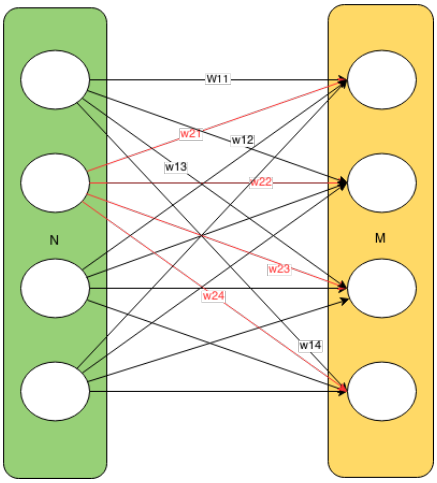


Figura 7.2: Dense connected layer

7.3 Input

The input is a dense layer with 22 neurons, expecting as input the normalized metrics. As activation function **SeLU** will be used.

7.4 Hidden layer

A single hidden dense hidden layer with 22 neurons will be used, with **SeLU** as activation function.

7.5 Drop out

For the small dimension of the network, no drop out will probably be necessary.

7.6 Output

The output layer of the network is a dense layer with one neuron and a **sigmoid** as activation function. When **active** models the positive class, and when **inactive** models the negative class.

$$\sigma(x) = \text{sigmoid}(x) = \frac{e^x}{e^x + 1}$$

Figura 7.3: Sigmoid

7.7 Weight initialization

Since input values are not from any particular distribution or hold properties such as $\mathbb{E}(X) = 0$ (even though the normalized data have null mean) or $\text{Var}(X) = 1$ (in some metrics mean and variance are far from these values) they do not suggest to use any specific distribution.

The codomain values from the activation functions, being SeLU for most of the network, tend to hold the properties of $\mathbb{E}(X) = 0$ and $\text{Var}(X) = 1$ (http://arxiv.org/abs/1706.02515).

For these properties weight will be initialized by extracting them from a Gaussian with $\mathbb{E}(X) = 0$ and $\text{Var}(X) = 1$.

7.8 Batch Size

Since the positive class is extremely smaller of the negative class a large batch size will be necessary: 4096 data points per batch allow for a probability that at least one of the data points is positive in the training set of the ≈ 0.77.

7.9 Loss function

Since the task assigned to the network is a binary classification the loss function will be the **cross entropy**.

7.10 Update policy

As update policy we are going to use a form of gradient **back propagation** with **Adam** optimizer (9).

7.11 Mathematical representation

Since the network is extremely simple a short mathematical representation is possible:

$$\text{prediction}(\underline{\boldsymbol{x}}) = \sigma(\boldsymbol{W}_{output} \cdot \text{SeLU}(\boldsymbol{W}_{hidden} \cdot \text{SeLU}(\boldsymbol{W}_{input} \cdot \underline{\boldsymbol{x}})))$$

Figura 7.4: Mathematical model

7.12 Network model representation

Graphical model of the network.

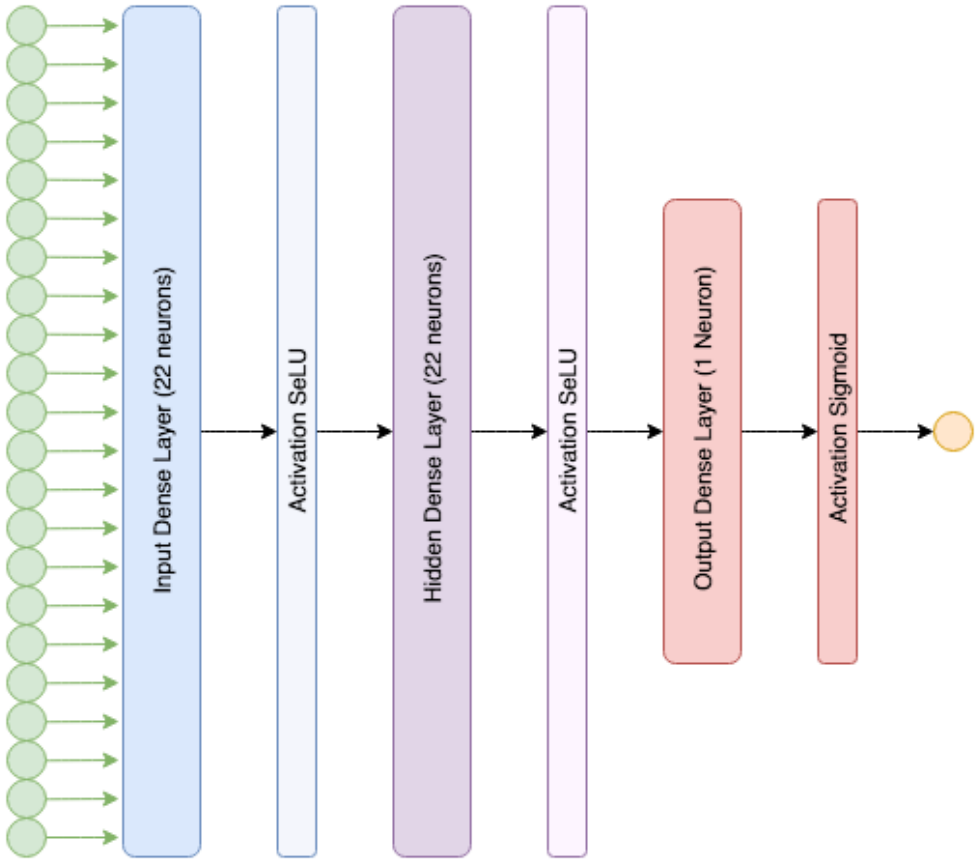


Figura 7.5: Model of the network

8 Results

These are the results obtained after training the network for 100 epochs.

8.1 Accuracy and Loss

Metric	Training	Testing
Accuracy	0.9998970845374103	0.9995267641182038
Loss	0.0004865645350134752	0.0018388369186131115

Tabella 8.1: Accuracy and Loss after 100 epochs

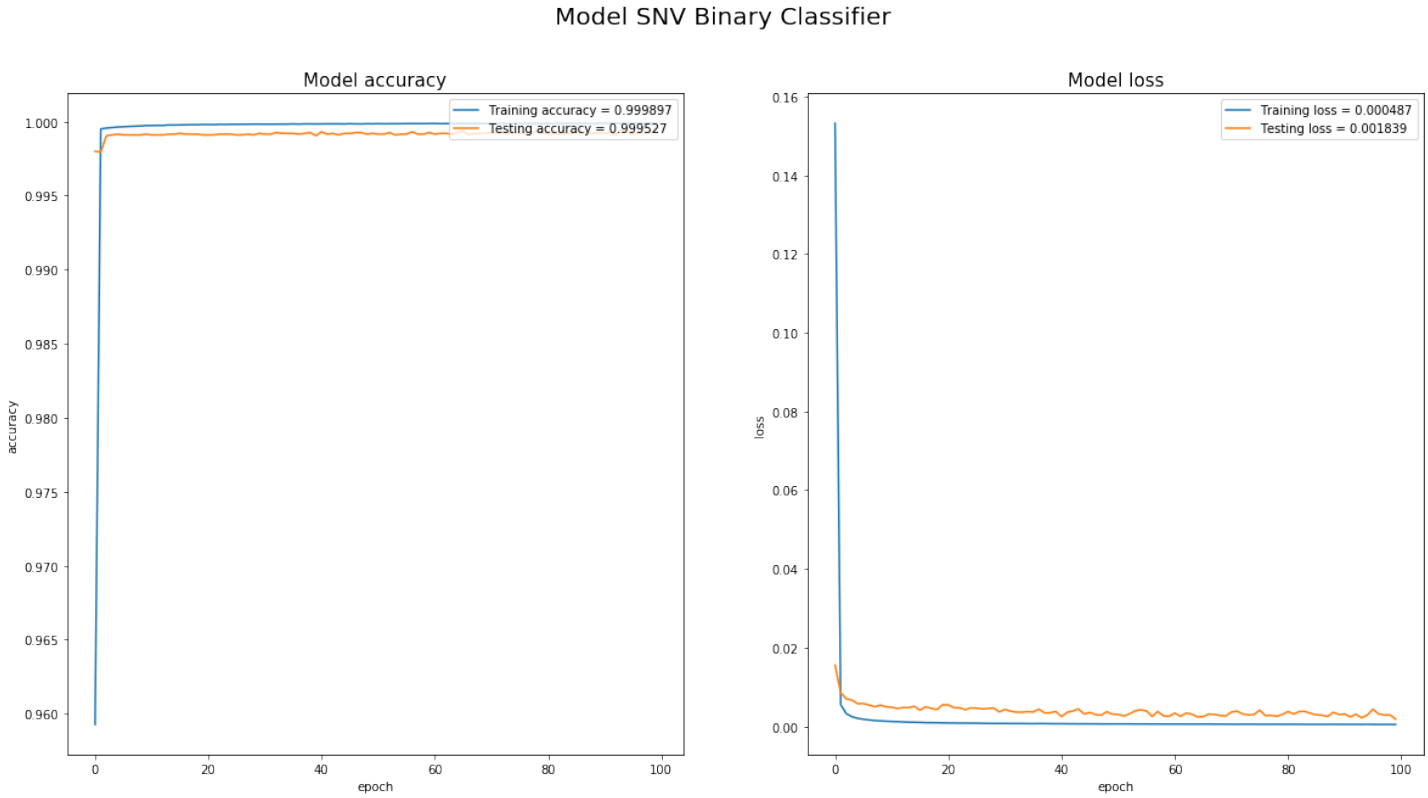


Figura 8.1: Accuracy and Loss after 100 epochs

8.2 ROC, AUROC, PRC and AUPRC

Metric	Training	Testing
AUROC	0.9982423662593878	0.9944672779007271
AUPRC	0.8780726019818784	0.9337718051372246

Tabella 8.2: Accuracy and Loss after 100 epochs

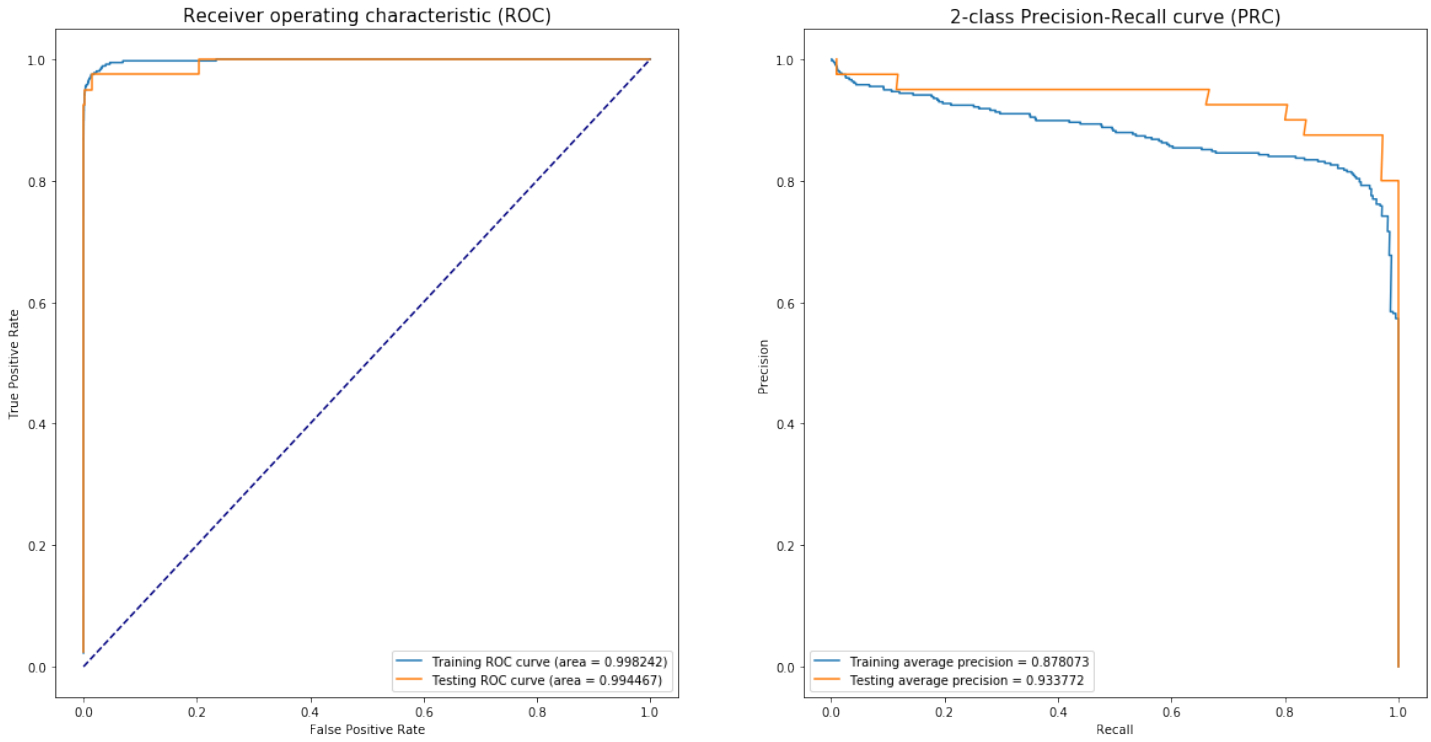


Figura 8.2: ROC, AUROC, PRC and AUPRC after 100 epochs

8.3 Confusion matrices

8.3.1 Training confusion matrix

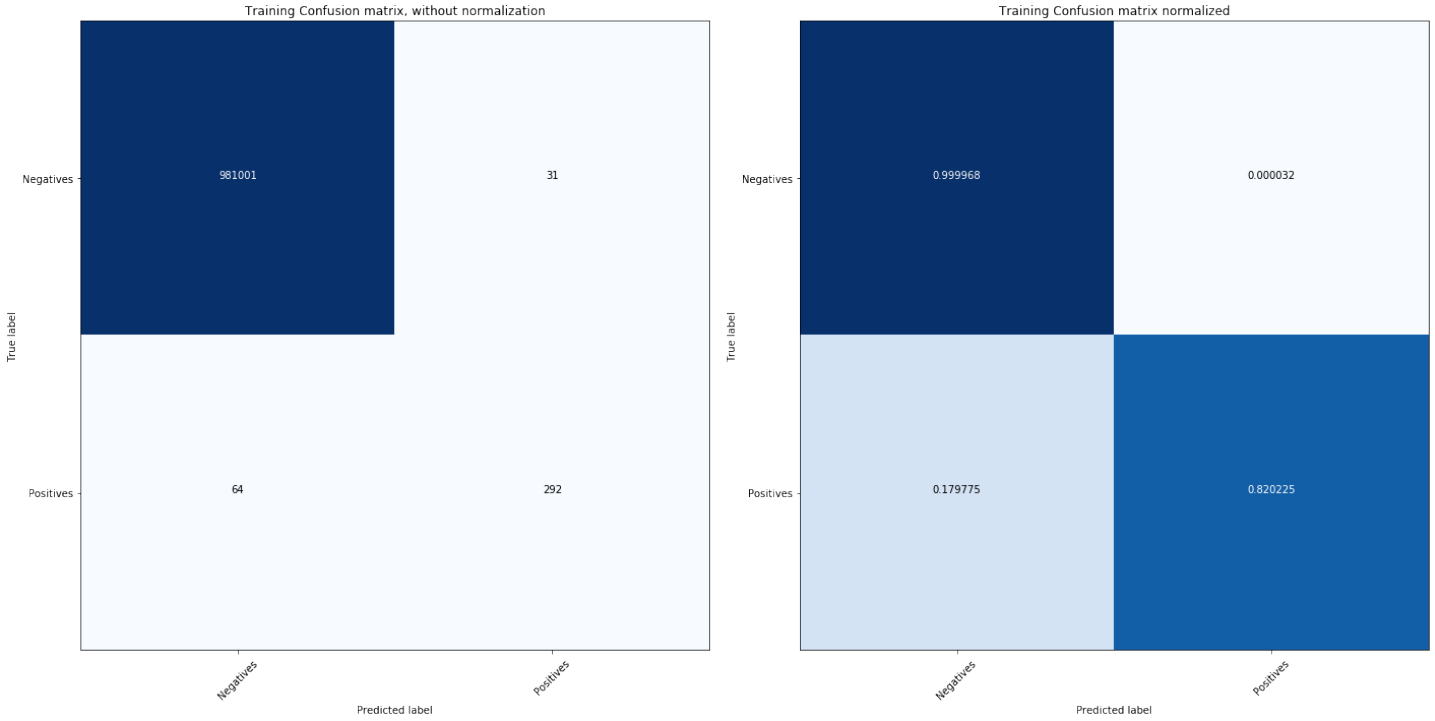


Figura 8.3: Training confusion matrix after 100 epochs

8.3.2 Testing confusion matrix

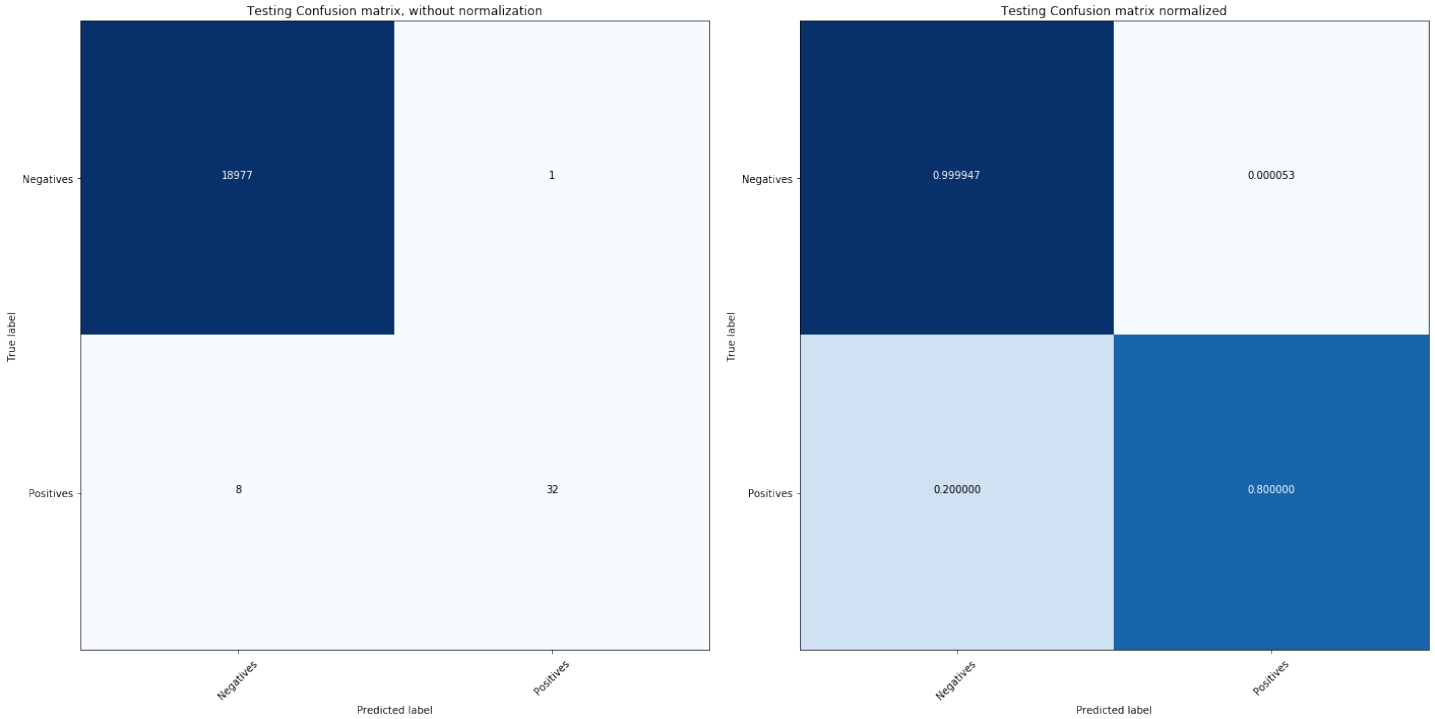


Figura 8.4: Testing confusion matrix after 100 epochs

8.4 Prediction speed

The network classifies a data point in $375\mu s \pm 108\mu s$.

This value was computer on a machine with the following carattestics:

Computer Specification	
Model Name	MacBook Pro
Processor Name	Intel Core i7
Processor Speed	2.3 GHz
Number of Processors	1
Total Number of Cores	4
L2 Cache (per Core)	256 KB
L3 Cache	6 MB
Memory	16 GB

Parte III

References

9 References

Adam Adam: A Method for Stochastic Optimization: https://arxiv.org/abs/1412.6980

SeLU Self-Normalizing Neural Networks: https://arxiv.org/abs/1706.02515