Supplemental Information for *Human methylome variation across Infinium 450K data on the Gene Expression Omnibus*

Sean K. Maden Reid F. Thompson Kasper D. Hansen Abhinav Nellore

Contents

Overview

This document contains supplemental information for the manuscript *Human methylome variation across Infinium 450K data on the Gene Expression Omnibus*, including details about data access, aggregation, and analyses, with links to specific scripts where applicable.

Resources summary

Key resources and their descriptions are as follows:

- `recountmethylation` – Bioconductor package to access, query, and analyze full database compilations and sample metadata. http://bioconductor.org/packages/devel/bioc/html/recountmethylation.html

- recountmethylationManuscriptSupplement – GitHub repo for manuscript supplemental files, scripts, and data. https://github.com/metamaden/recountmethylationManuscriptSupplement
- recount.bio/data – Location of large supplemental data files. https://recount.bio/data/recountmethylation_manuscript_supplement/
- `recount-methylation-server` – Server software for identifying, downloading, and managing IDATs and SOFT files. https://github.com/metamaden/recount-methylation-server
- `rmpipipeline` – GitHub repo with resources for processing GEO IDATs and SOFT files. https://github.com/metamaden/rmpipeline

Gene Expression Omnibus (GEO) data

DNA methylation (DNAm) array samples were identified as published to the Gene Expression Omnibus (GEO) and available in the GEO Data Sets database as of March 31, 2019.

GEO queries and data summaries

The Entrez Utilities software (v.10.9) was used to quantify DNAm array sample availability by year and platform for 3 major Illumina BeadArray platforms (HM450K, HM27K, and EPIC/HM850K.

Samples and studies were identified for download using the script https://github.com/metamaden/recount-methylation-server/blob/master/src/edirect_query.py.

Data availability by year was determined using the script https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/python/eqplot.py.

Plots for figures 1A and S1 were generated using the scripts https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/figures/fig1a.R and https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/figures/figS1.R.

Data acquisition

We used the Python programming language to develop a download management system to handle and version file downloads from GEO. We used this job management system to obtain GSM IDATs, GSE SOFT files, and other data using batch queries to the GEO Data Sets repository.

IDATs and SOFT files were acquired using the script https://github.com/metamaden/recount-methylation-server/blob/master/src/dl.py.

Processing DNAm data from IDATs

Signals were read from sample IDATs into an R session using the `minfi` package (v.1.29.3, Aryee et al. (2014)). DNAm assay data were read from IDATs using the script https://github.com/metamaden/rmpipeline/blob/master/R/rmpipeline.R.

Learning sample metadata from SOFT files

Metadata preprocessing and postprocessing was performed, resulting in the samples metadata table S1 used for manuscript analyses (Table S1).

Extracting sample/GSM metadata from GSE SOFT files in JSON format

GSE SOFT files were converted to JSON format using the script https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/metadata/soft2json.R. JSON files were then filtered using the script https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/metadata/jsonfilt.R. The filtered JSON data were then stored as a list with https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/metadata/make_gse_annolist.R, and the list was coerced into the preprocessed metadata object using https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/metadata/gseanno_manualharmonize.R.

Postprocessing mined sample metadata under common variables

We used term mapping with regular expressions to create the `disease` and `tissue` columns in the postprocessed metadata (Table S1). The preprocessed metadata table (see above) was postprocessed using the script https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/metadata/md_postprocessing.R

This script used regular expressions mapping, including inclusive and exclusive mapping logic, to learn new terms and annotations. This mapped mined metadata into a regulated series of terms similar to those implemented in the marmal-aid resource. In summary, labels have general labels applied automatically and separated by semicolons, and resulting labels were all lowercase, where underscores replace spaces for applicable labels. For example, a label of "Peripheral blood" is converted into "peripheral_blood;blood".

Predicting sample types from filtered sample JSON files

Filtered GSM data in JSON format were derived from GSE SOFT files (see above). Filtering removed the GSE-specific data, including experiment title and methods text. This step helped distinguish metadata between samples with different characteristics. For example, the filtered JSON file for sample GSM937258 was:

```
[
{
  "!Sample_characteristics_ch1": "tissue site: Bone",
  "!Sample_characteristics_ch1.1": "tissue type: Tumor",
  "!Sample_characteristics_ch1.2": "sample id (short form for plotting): 24-6b",
  "!Sample_characteristics_ch1.3": "description: Xiphoid Met",
  "!Sample_source_name_ch1": "Prostate cancer metastasis",
  "!Sample_title": "Prostate cancer metastasis 16032"
}
]
```

We ran the MetaSRA-pipeline software on the filtered JSON files (Bernstein et al. (2017)). For this, we forked the original software, and made light modifications so that it ran successfully using Python 3. This fork is available at https://github.com/metamaden/MetaSRA-pipeline.

MetaSRA-pipeline mapped ontology term labels and made sample type predictions (Figure S2). We appended the most likely sample types and their likelihoods under the "sampletype" column in the postprocessed metadata. For applicable samples, cell line annotations were appended from the Cellosaurus resource.

DNAm model-based predictions for age, sex, and blood cell types

Model-based predictions of sex, age, and cell type fractions were obtained from DNAm data as described in https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/metadata/metadata_model_predictions.R.

Age predictions were obtained by running the `agep` function from the `wateRmelon` (v.1.28.0) package on noob-normalized Beta-value DNAm, which used the method described in Horvath 2013. Sex predictions were obtained by running the `getSex` function from `minfi` on red and green channel data. Cell fraction predictions were obtained by running the `estimateCellCounts` function from `minfi` using red and green channel data (Houseman et al. (2012)).

The `getSex` and `estimateCellCounts` functions perform light preprocessing that may be influenced by input sample order. We found that repeated calculations with randomized sample ordering showed > 95% concordance for both variables (data not shown).

Detailed column decriptions for postprocessed/final metadata

1. `gsm` : GEO sample record ID
2. `gsm_title` : GEO sample record title
3. `gseid` : GEO study record ID
4. `disease` : Learned disease characteristics and study group labels
5. `tissue` : Learned tissue characteristics labels
6. `sampletype` : Most likely sample type from MetaSRA-pipeline

- `msraptype`: Sample label
- `msrapconf`: Prediction confidence
- `ccid`, `ccacc`, and `cccat`: Mapped cellosaurus id, accession, and attribute info where available/applicable

7. `arrayid_full`: Full array Sentrix ID (format: `Sentrix ID_chip coordinate`)
8. `basename`: Basename, or shared path, to sample IDAT files
9. `age`: Learned chronological age and age units.

- `valm`: Age value
- `unitm`: Age units

10. `predage`: DNAm-based age estimate, predicted using noob-normalized Beta-valuees and `agep()` from the `wateRmelon` package.
11. `sex`: Learned sex label (M = male, F = female).
12. `predsex` : DNAm-based sex, predicted from unnormalized red and green signal using `getSex()` from the `minfi` R package (M = male, F = female).
13. `predcell.CD8T` : DNAm-based CD8T cell fraction prediction from `estimateCellCounts()` in `minfi`.
14. `predcell.CD4T` : DNAm-based CD4T cell fraction prediction from `estimateCellCounts()` in `minfi`.
15. `predcell.NK` : DNAm-based Natural Killer cell prediction from `estimateCellCounts()` in `minfi`.
16. `predcell.Bcell` : DNAm-based Bcell fraction prediction from `estimateCellCounts()` in `minfi`.
17. `predcell.Mono` : DNAm-based Monocyte fraction prediction from `estimateCellCounts()` in `minfi`.
18. `predcell.Gran` : DNAm-based Granulocyte fraction prediction from `estimateCellCounts()` in `minfi`.

19. `storage` : Manually annotated storage procedure (`FFPE` for formalin-fixed paraffin embedded or `F` for fresh frozen).

Quality and summary metrics

We calculated quality signals for 17 BeadArray controls, methylated and unmethylated signals, and likely sample replicates by study, as described in the script https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/inst/scripts/tables/tableS2.R

BeadArray signals for 17 controls (summarized in Table S3) were calcuated using the script https://github.com/metamaden/recountmethylationManuscriptSupplement/blob/main/R/beadarray_cgctrlmetrics.R. Our method was developed in consultation with Illumina's platform documentation ((“Illumina Genome Studio Methylation Module V1.8” 2010), (“BeadArray Controls Reporter Software Guide” 2015)) and functions from the `ewastools` R package (Heiss and Just (2018)).

Sample genetic identities were calculated for GSE records using the `call_genotypes` and `check_snp_agreement` functions from the `ewastools` package (Heiss and Just (2018)), with a likelihood cutoff of 0.1. This method uses a probabilistic model high-frequency SNP data probed by the HM450K platform.

Statistical analyses

Summary statistics and statistical tests

Sample data were obtained, read, and analyzed programmatically using the R (v.4.0.0) and Python (v.3.7.0) languages in a CentOS 7 remote server environment. IDAT signals were read into `SummarizedExperiment` objects using the `minfi` package. Summary statistics were generated using base R functions. Statistical tests were performed using the `stats` (v.4.0.0) R package. Correlation tests used the Spearman method by setting `method = "spearman"` in the `cor.test` function. Analyses of variance (ANOVAs) were performed using the `anova` function. Label enrichments were tested using Binomial with the `binom.test` function, and T-tests used the `t.test` function. Principal component analyses (PCA) used the `prcomp` function. Unless noted otherwise, p-value adjustments used the Benjamini-Hotchberg method by setting `method = "BH"` in the `p.adjust` function. Plots used either base R, `ggplot2` (v.3.1.0), or `ComplexHeatmap` (v.1.99.5).

Autosomal DNAm principal component analyses (PCAs)

Approximate array-wide PCA was performed using noob-normalized Beta-values from autosomal probes and teh `prcomp` from the `stats` R package. We condensed autosomal DNAm across 35,360 samples into 1,000 hashed feature columns, then used this reduced representation (1,000 rows X 35,360 columns) in cluster analysis. Feature hashing, or the “hashing trick” (Weinberger et al. (2010)), was used as an intermediate dimensionality reduction step. The data files for each performed test are available at https://recount.bio/data/recountmethylation_manuscript_supplement/data/.

To perform feature hashing, we used the following script in Python 3, where `target_dim=1000` specifies the target hashed feature dimension:

```
def feature_hash(arr, target_dim=1000):
    low_d_rep = [0 for _ in range(target_dim)]
    for i, el in enumerate(arr):
        hashed = mmh3.hash(str(i))
        if hashed > 0:
            low_d_rep[hashed % target_dim] += arr[i]
        else:
            low_d_rep[hashed % target_dim] -= arr[i]
    return low_d_rep
```

Predicted and mined ages concordances

For mined and predicted ages, ANOVAs were used to calculate covariate variance percentages, and p-values, for multivariate models of predicted/epigenetic age consisting of chronological age, GSE ID, cancer status, and predicted sample type. Predicted age and chronological age were regressed to calculate R-squared values and correlated using Spearman's test. These were plotted with the regression model using `ggplot2` (Figure 1b). See example 1 in the `data_analyses` vignette of the `recountmethylation` package for details.

Works cited

Aryee, Martin J., Andrew E. Jaffe, Hector Corrada-Bravo, Christine Ladd-Acosta, Andrew P. Feinberg, Kasper D. Hansen, and Rafael A. Irizarry. 2014. "Minfi: A Flexible and Comprehensive Bioconductor Package for the Analysis of Infinium DNA Methylation Microarrays." *Bioinformatics* 30 (10): 1363–9. https://doi.org/10.1093/bioinformatics/btu049.

"BeadArray Controls Reporter Software Guide." 2015, October. https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/infinium_assays/infinium_hd_methylation/beadarray-controls-reporter-user-guide-1000000004009-00.pdf.

Bernstein, Matthew N., AnHai Doan, Colin N. Dewey, and Jonathan Wren. 2017. "MetaSRA: Normalized Human Sample-Specific Metadata for the Sequence Read Archive." *Bioinformatics* 33 (18): 2914–23. https://doi.org/10.1093/bioinformatics/btx334.

Heiss, Jonathan A., and Allan C. Just. 2018. "Identifying Mislabeled and Contaminated DNA Methylation Microarray Data: An Extended Quality Control Toolset with Examples from GEO." *Clinical Epigenetics* 10 (June). https://doi.org/10.1186/s13148-018-0504-1.

Houseman, Eugene A., William P. Accomando, Devin C. Koestler, Brock C. Christensen, Carmen J. Marsit, Heather H. Nelson, John K. Wiencke, and Karl T. Kelsey. 2012. "DNA Methylation Arrays as Surrogate Measures of Cell Mixture Distribution." *BMC Bioinformatics* 13 (1): 86. https://doi.org/10.1186/1471-2105-13-86.

"Illumina Genome Studio Methylation Module V1.8." 2010. Illumina. https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/genomestudio/genomestudio-2011-1/genomestudio-methylation-v1-8-user-guide-11319130-b.pdf.

Weinberger, Kilian, Anirban Dasgupta, Josh Attenberg, John Langford, and Alex Smola. 2010. "Feature Hashing for Large Scale Multitask Learning." *arXiv:0902.2206 [Cs]*, February. http://arxiv.org/abs/0902.2206.