Chapter 9

Genome-Wide Analysis of DNA Methylation in *Arabidopsis* Using MeDIP-Chip

Sandra Cortijo, René Wardenaar, Maria Colomé-Tatché, Frank Johannes, and Vincent Colot

Abstract

DNA methylation is an epigenetic mark that is essential for preserving genome integrity and normal development in plants and mammals. Although this modification may serve a variety of purposes, it is best known for its role in stable transcriptional silencing of transposable elements and epigenetic regulation of some genes. In addition, it is increasingly recognized that alterations in DNA methylation patterns can sometimes be inherited across multiple generations and thus are a source of heritable phenotypic variation that is independent of any DNA sequence changes. With the advent of genomics, it is now possible to analyze DNA methylation genome-wide with high precision, which is a prerequisite for understanding fully the various functions and phenotypic impact of this modification. Indeed, several so-called epigenomic mapping methods have been developed for the analysis of DNA methylation. Among these, immunoprecipitation of methylated DNA followed by hybridization to genome tiling arrays (MeDIP-chip) arguably offers a reasonable compromise between cost, ease of implementation, and sensitivity to date. Here we describe the application of this method, from DNA extraction to data analysis, to the study of DNA methylation genome-wide in *Arabidopsis*.

Key words DNA methylation, 5-Methylcytosine (5mC), MeDIP, Tiling array, Epigenetic variation

1 Introduction

In eukaryotes, DNA methylation almost exclusively affects cytosines (5-methylcytosines). Once established, this modification can be maintained over numerous cell divisions and even across generations in some instances. However, it remains unclear to what extent differences in DNA methylation can be stably inherited and this question is the subject of intense studies. This is especially true in *Arabidopsis*, where epigenetic recombinant inbred lines (epiRILs) have been derived from parents with few differences in DNA sequence but contrasted DNA methylation profiles [1, 2]. One such population of epiRILs has been epigenotyped [3] in order to assess the stability of parental DNA methylation differences and

Charles Spillane and Peter C. McKeown (eds.), *Plant Epigenetics and Epigenomics: Methods and Protocols*,
Methods in Molecular Biology, vol. 1112, DOI 10.1007/978-1-62703-773-0_9, © Springer Science+Business Media New York 2014

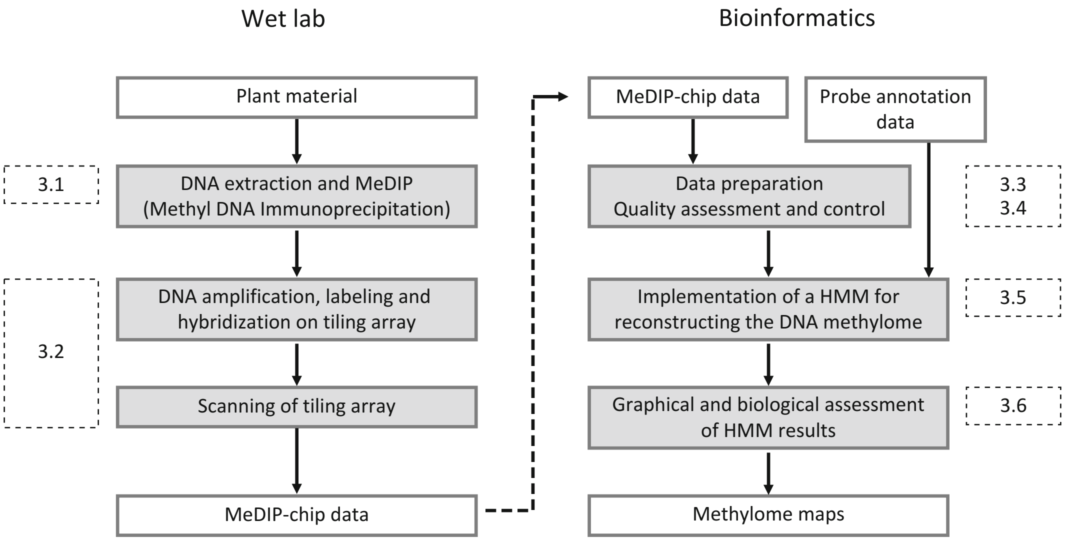
Wet lab
Bioinformatics
Plant material
3.1
DNA extraction and MeDIP (Methyl DNA Immunoprecipitation)
3.2
DNA amplification, labeling and hybridization on tiling array
Scanning of tiling array
MeDIP-chip data
MeDIP-chip data
Probe annotation data
Data preparation Quality assessment and control
3.3
3.4
Implementation of a HMM for reconstructing the DNA methylome
3.5
Graphical and biological assessment of HMM results
3.6
Methylome maps

Fig. 1 Flowchart for the reconstruction of methylome maps (Subheading 3)

their impact on several complex traits. Here, we describe the methyl DNA immunoprecipitation (MeDIP)-chip protocol used to reconstruct the DNA methylome maps, starting from the extraction of DNA to analysis of the hybridization data using Hidden Markov Models (HMM; *see* flow chart in Fig. 1). Subheading 2 lists the materials needed for the "wet" part as well as the software and data used for analysis. Subheading 3 describes step by step the MeDIP-chip experiment (Subheadings 3.1 and 3.2) and the analysis of hybridization data, starting from data preparation (Subheading 3.3), then quality assessment and control (Subheading 3.4), implementation of a HMM for reconstructing DNA methylome maps (Subheading 3.5) and graphical and biological assessment of HMM results (Subheading 3.6).

2 Materials

2.1 DNA Extraction and Methyl DNA Immunoprecipitation

1. DNA extraction with DNeasy plant Maxi kit (Qiagen, Catalogue N° 68163).
2. 1.5 mL Siliconized tubes: Clear-view™ Snap-Cap microtubes, siliconized (Sigma, Catalogue N° T4816-250EA).
3. Sonicator: Bioruptor (Diagenode, Catalogue N° UCD-200).
4. Buffer 1: 13.3 mM Tris–HCl pH 7.5, 667 mM NaCl, 1.3 mM EDTA.
5. Monoclonal antibody against 5mC (Diagenode, Catalogue N° MAb-006-500).
6. Rotating wheel.

7. Magnetic beads: M280 Dynabeads (Invitrogen, Catalogue # 112-01D).
8. Buffer 2: 10 mM Tris–HCl pH 7.5, 500 mM NaCl, 1 mM EDTA.
9. Buffer 3: 30 mM Tris–HCl pH 8.0.
10. Proteinase K: 20 μg/μL (NEB, Catalogue N° P8102S).
11. Phenol/chloroform/IAA (25:24:1, pH 8.0) and Chloroform/IAA: (24:1).
12. Glycogen azure: 20 μg/μL, resuspended in water (Sigma, Catalogue N° G5510-1G).
13. NaOAc: 3 M, pH 5.2.
14. MinElute Reaction Cleanup Kit (Qiagen, Catalogue N° 28204).
15. PicoGreen: Quant-it PicoGreen dsDNA reagent (Invitrogen, Catalogue # P7581) diluted to 0.5 % in TE, pH 8.

2.2 DNA Amplification, Labeling, and Hybridization on Tiling Array

1. WGA2 kit (Sigma, Catalogue # WGA2-50RXN).
2. QIAquick PCR Purification Kit (Qiagen, Catalogue N° 28104).
3. Dual Color DNA labeling kit (NimbleGen, Catalogue N° 06370250001).
4. Hybridization and wash buffer kits (NimbleGen, Catalogue N° 05583683001 and 05584507001).
5. Scanner: High-Resolution (2 μm) Microarray Scanner (Agilent, Catalogue N° G2565CA).
6. NimbleScan software (NimbleGen).

2.3 Software Requirements

This protocol requires R for the analysis of the MeDIP-chip data. R is a command line-based software environment for statistical computing and graphics. It can be freely downloaded at http://www.r-project.org and installed on all three main operating systems (Windows, Unix/Linux, and Mac). Instructions about installation and tutorials can be obtained from the same website. R is extensively used among biostaticians due to the availability of statistical packages for the analysis of a broad spectrum of biological data. In addition to R, we also recommend downloading a text editor with syntax highlighting (e.g., Notepad++). Programming mistakes are more easily detected when using a text editor. All the code lines and functions are highlighted throughout the chapter in courier font. The HMM is implemented in C++. An electronic version of the R code presented in this chapter and the HMM software are freely available at the following URL: http://www.johanneslab.org/publications. This chapter does not show the code for generating the figures. This code can, however, be downloaded from the same URL.

Table 1
Format methylation data

PROBE_ID	REP1_INPUT_RED	REP2_INPUT_RED	REP3_INPUT_RED
CHR01FS000000061	778.53	2534.67	1033.31
CHR01FS000000212	2366.51	2756.02	1333.69
CHR01FS000000382	4028.27	7776.75	3201.88
CHR01FS000000507	13685.61	15014.29	8556.37
CHR01FS000000707	1565.45	2626.51	1187.04

2.4 Dataset

The protocol was implemented for the efficient and cost-effective genome-wide study of DNA methylation of a large number of *Arabidopsis* lines. The dataset used to illustrate this protocol can be downloaded from the above URL and consists of six files that contain the measured signal intensities (IP and INPUT) for one wild type line (Columbia accession, Col-0), probe annotation, conservation scores for probes, and an example of an array with a hybridization artefact.

2.4.1 Methylation Data

The methylation data should be tab-delimited and have the format shown in Table 1. The first column of the file should contain the probe identifier and the remaining column (or columns when replicates are available) should contain the measured probe intensities. The IP and INPUT files should have the same tab-delimited format.

2.4.2 Hybridization Artefact Data

For illustrative purposes, we also show an example of a hybridization artefact (Fig. 2a). This file should also be tab-delimited and have the format shown in Table 2. The first column should again contain the probe identifier, the second and third column should contain the location of the probe on the array (*x* and *y* position on the array) and the fourth column (PM) should contain the measured probe intensity (IP or INPUT signal).

2.4.3 Conservation Score Data

The conservation score of a probe indicates the uniqueness of the probe sequence (not all probe sequences are unique). This score was obtained by performing a BLAST search. Scores are percentage of identity with the second best hit (the first hit is the location in the genome for which the probe was designed). Probes can be visualized at http://epigara.biologie.ens.fr/index.html. The conservation score data should have the tab-delimited format shown in Table 3.

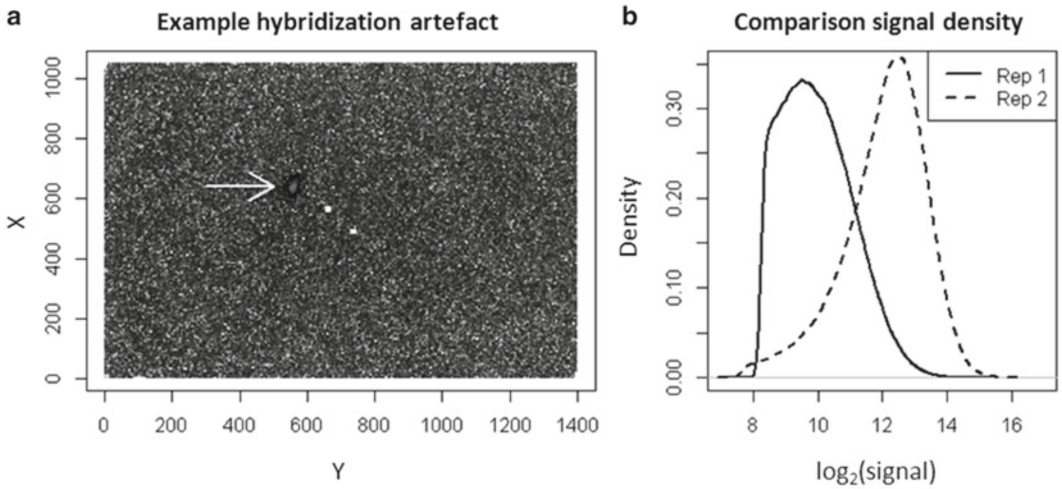

Fig. 2 Quality of the overall hybridization experiment. (**a**) The *arrow* points to an unwanted spatial artefact on the tiling array. One could consider excluding the relevant probes or discarding the tiling array entirely. (**b**) Shown is the signal density distribution of the Cy3 INPUT channel for two replicates. One of the individuals (*solid line*) shows a steep increase in the lower signal range suggesting that an insufficient amount of DNA was hybridized to the tiling array. The signal distribution of the other replicate (*dashed line*) is normal. The bulk of the data is located in the center of the detection range indicating that the right amount of DNA was hybridized to the tiling array

Table 2
Format hybridization artefact data

PROBE_ID	X	Y	PM
CHR01FS000000061	327	1335	3219.96
CHR01FS000000212	191	1257	4840.31
CHR01FS000000382	826	34	8668.02
CHR01FS000000507	731	529	19781.76
CHR01FS000000707	624	562	1195.29

Table 3
Format conservation score data

PROBE_ID	Score
CHR01FS000000061	73
CHR01FS000000212	56
CHR01FS000000382	64
CHR01FS000000507	62
CHR01FS000000707	74

Table 4
Format annotation data

PROBE_ID
CHR01FS000004351
CHR01FS000005311
CHR01FS000007129
CHR01FS000007479
CHR01FS000007814

2.4.4 Annotation Data

The annotation files contain the probe identifiers of probes that are located within introns of protein-coding genes or transposons. The annotation data should only contain one column with probe identifiers as shown in Table 4.

3 Methods

3.1 DNA Extraction and MeDIP

1. Extract DNA from plant material (1–2 g fresh weight, we use aerial parts of 3-week-old plants grown under long day conditions) with Qiagen DNeasy plant Maxi kit. 1.8 μg of DNA is needed for this protocol (includes sonication test and INPUT and IP fractions).
2. Quantify DNA and place 1.8 μg in a final volume of 180 μL (complete with water if necessary) in 1.5 mL siliconized Eppendorf tubes. Set aside 2 μL (corresponding to 20 ng of DNA) for sonication control. Sonicate the remaining 178 μL using seven cycles of 30 s ON/30 s OFF. Note that all six positions within the sonicator need to be filled, with an equal volume of water (178 μL) put in each tube. Place all six tubes in an ice bucket and add ice to the sonicator bath to cool it off. Repeat sonication once (14 cycles in total). Keep 13 μL to test sonication (sonicated fraction).
3. Run non-sonicated (2 μL) and sonicated (13 μL) samples side by side in 1.5 % 1× TAE gel. A smear should be visible between 100 and 600 bp (with maximum intensity around 300 bp) after sonication.
4. Keep 15 μL to serve as INPUT (150 ng). Use the remaining 150 μL of sonicated DNA (1.5 μg) for IP.
5. Add 450 μL of buffer 1 to IP fraction (total volume of 600 μL). Incubate 10 min at 95 °C to denature DNA (this is critical as the antibody only recognizes 5mC on single-stranded DNA!) and let sit on ice for 2 min. Add 5 μL of 1 μg/μL anti-5mC antibody to denatured IP fraction. Close

tubes, wrap with parafilm (siliconized tubes tend to leak) and incubate overnight at 4 °C with gentle agitation (we use a rotating wheel, with a 45° inclination, 8 rpm).

6. Use 40 μL of magnetic beads per MeDIP. Prepare a tube with the total amount of beads required for the number of MeDIP performed. Wash the beads three times with 1 mL of buffer 2 (*see* **Note 1**) and resuspend one last time with buffer 2 in the starting volume. Put 40 μL of washed beads, making sure that they are well resuspended by pipetting up and down the slurry several times, into each MeDIP tube. Put on rotating wheel for 4 h at 4 °C with gentle agitation (45° inclination, 8 rpm).
7. Put IP samples on the Dynabeads rack (magnetic rack). Collect supernatant in a new 2 mL Eppendorf tube (supernatant fraction). Add 300 μL of buffer 2 to IP tube. Agitate briefly by hand, and place for 10 min at room temperature on the rotating wheel with gentle agitation (45° inclination, 8 rpm). Put back on the Dynabeads rack and add first wash to supernatant fraction. Perform three more washes, each time with 600 μL of buffer 2. Discard washes.
8. Add 300 μL of buffer 3 to the IP pellet after last wash and transfer IP and supernatant fractions to 1.5 mL and 2 mL tubes, respectively (*see* **Note 2**). Add 7 μL of Proteinase K to elute. Incubate 1 h at 42 °C, with occasional shaking.
9. Add one volume of phenol/chloroform/IAA to the IP and supernatant fractions (300 and 900 μL, respectively). Vortex and centrifuge 5 min at 14,000 × *g* at room temperature. Place aqueous phase (top phase) in a new tube. Add one volume of chloroform/IAA to aqueous phase. Vortex and centrifuge 5 min at 14,000 × *g* at room temperature. Place aqueous phase in a new tube.
10. To precipitate DNA, add 1 μL of glycogen azure, 1:10 volume of NaOAc, and one volume of isopropanol to the IP and supernatant fractions (30 and 90 μL for NaOAc and 300 and 900 μL for isopropanol in IP and supernatant fraction, respectively). Vortex between addition of each component. Keep at –20°C for at least 1 h or overnight. Centrifuge 30 min at room temperature at max speed (>13,000 × *g*). Discard supernatant and add 500 μL of ethanol 70 %. Mix and centrifuge for 20 min at room temperature at max speed (>13,000 × *g*). Discard the supernatant and dry DNA pellets by leaving the tubes open on the bench for ~30 min. Resuspend all DNA pellets in 40 μL of TE, pH 8.0, and add 25 μL to INPUT fraction.
11. Perform quantitative PCR on the three fractions (IP, supernatant, and INPUT) with known positive and negative controls before proceeding with purification, labeling, and hybridization

to tiling array. Note that for wild type *Arabidopsis* (Columbia accession), approximately 10–20 % of the genome should be immunoprecipitated with the anti-5mC antibody for DNA extracted from aerial or root parts.

12. DNA should be cleaned one last time using the MinElute kit (*see* **Note 3**). Expect 30 % loss of DNA.
13. DNA concentration is checked with Nanodrop 3300. Add 2 μL of diluted PicoGreen at 0.5 % to 2 μL of DNA and quantify this mix using function "dsDNA PicoGreen® dye" in "Nucleic Acid Quantitation" (*see* **Note 4**).

3.2 DNA Amplification, Labeling, and Hybridization on Tiling Array

1. Use 10 ng of IP and 50 ng of INPUT fractions for amplification with the WGA2 kit. Start from the "Library preparation" step of the protocol, as there is no need for the DNA fragmentation step.
2. Purification of the amplification products is carried out using QIAquick PCR Purification Kit. Quantify and run in a 1.5 % agarose TAE 1× gel. This should produce a smear corresponding to the sonication smear (between 100 and 600 bp). Final yield fluctuates between 3 and 6 μg.
3. DNA labeling is carried out using the Dual Color DNA labeling kit, using 1 μg of amplified IP and INPUT DNA. Resuspend labeled DNA in 20 μL of water and quantify it, together with Cy3 and Cy5 using the "microarray function" of the Nanodrop 2000. One should expect 10–20 μg of DNA after labeling and 200–400 pmol of incorporated dye. Repeat labeling if DNA yield or incorporation levels are less than 5 μg or 100 pmol, respectively (*see* **Note 5**).
4. Differential hybridization is carried out using a NimbleGen 3x720K tiling array design (three identical chambers, design available on request) and following the manufacturer's instructions. Use 4 μg of each of the two labeled DNA samples (IP and INPUT) per chamber. Hybridization is in dye-swap (IP in red and INPUT in green for the first chamber and vice versa for the second chamber).
5. After washing, the NimbleGen 3x720K tiling array is scanned using a High-Resolution (2 μm) Microarray Scanner (Agilent). It is preferable to scan each chamber independently.
6. Grid alignment and pair files extraction are made using the NimbleScan software and following the manufacturer's instructions.

3.3 Data Preparation

Following the "wet lab" part, one is confronted with a substantial amount of data ready to be analyzed. Before we show how this can be achieved, we detail several data preparation steps. The following commands are used to import the data in the R workspace. The command `setwd()` sets the working directory, such that there is no need to define the complete pathname of your files. The command `head()` shows the first lines of the file.

```
> setwd("D:\\reconstruction_methylome_maps")
>  input_wt  <-  read.table(file="input_wild_type.
txt",
+ header=TRUE,sep="\t")
> ip_wt    <- read.table(file="ip_wild_type.txt",
+ header=TRUE,sep="\t")
>
> head(input_wt)
```

```
   PROBE_ID          REP1_INPUT_RED    REP2_INPUT_RED    REP3_INPUT_RED
1  CHR01FS000000061  778.53            2534.67           1033.31
2  CHR01FS000000212  2366.51           2756.02           1333.69
3  CHR01FS000000382  4028.27           7776.75           3201.88
4  CHR01FS000000507  13685.61          15014.29          8556.37
5  CHR01FS000000707  1565.45           2626.51           1187.04
6  CHR01FS000000827  5939.94           7285.02           3212.73

   REP1_INPUT_GREEN  REP2_INPUT_GREEN  REP3_INPUT_GREEN
1  408.61            2038.57           818.98
2  712.76            2019.65           649.84
3  1350.67           5406.18           2090.43
4  2980.53           9570.41           5614.20
5  611.33            2405.53           460.63
6  1162.24           4555.31           2311.96
```

The IP and INPUT data have the same format; hence, there is no need to show the first lines of both files. We convert the data to a logarithmic scale using the following commands:

```
> log2_ip_wt                <- log2(ip_wt[,2:7])
> log2_ip_wt                <-  data.frame(ip_wt[,1],
                            log2_ip_wt)
> names(log2_ip_wt)[1]      <- "PROBE_ID"
 >
> log2_input_wt             <- log2(input_wt[,2:7])
> log2_input_wt             <- data.frame(input_wt[,1],
                            log2_input_wt)
> names(log2_input_wt)[1]   <- "PROBE_ID"
```

After log transformation, the datasets will have the same format only the signal intensities will be log transformed. In order to determine enrichment for DNA methylation, one has to calculate the intensity ratio of the IP and INPUT signal ($\log_2$ ratios). The following commands are used to calculate the intensity ratios. The dye-swapped replicates have been treated separately in this case (i.e., IP_{green} and $INPUT_{red}$ and vice versa). The IP and INPUT signals have also been averaged.

```
> wt_ip_green          <- (log2_ip_wt[,5]+log2_ip_wt
                          [,6]+log2_ip_wt[,7])/3
> wt_input_red         <-   (log2_input_wt[,2]+log2_
                          input_wt[,3]+
+ log2_input_wt[,4])/3
> wt_green_red         <- wt_ip_green-wt_input_red
> wt_green_red         <-                      data.
                         frame(log2_ip_wt[,1],wt_
                         green_red)
> names(wt_green_red) <- c("PROBE_ID","GREEN_RED")
>
> wt_ip_red            <- (log2_ip_wt[,2]+log2_ip_wt[,3]
                          +log2_ip_wt[,4])/3
> wt_input_green       <-          (log2_input_wt[,5]
                         +log2_input_wt[,6]+
+ log2_input_wt[,7])/3
> wt_red_green         <- wt_ip_red-wt_input_green
> wt_red_green         <-  data.frame(log2_ip_wt[,1],
                         wt_red_green)
> names(wt_red_green)  <- c("PROBE_ID","RED_GREEN")
```

After the calculation of the intensity ratios, the dye-swap signals can be calculated using the following code:

```
> wt_dye_swap          <-   (wt_green_red[,2]+wt_red_
                          green[,2])/2
> wt_dye_swap          <- data.frame(wt_green_red[,1],
                          wt_dye_swap)
> names(wt_dye_swap)   <- c("PROBE_ID","DYE_SWAP")
```

The dye-swap should account for possible dye bias in experiments. The data is now ready for subsequent analysis steps.

3.4 Quality Assessment and Control

Prior to array data analysis, we conduct detailed quality checks of each tiling array experiment. This quality assessment is necessary to ensure biologically meaningful results later on. If the data contains systematic hybridization artefacts or technical variation beyond a certain acceptable level, it is advisable to remove or to repeat the bad sample. We distinguish between two levels of quality assessment. The first level relates to the quality of the overall hybridization experiment and the second level to the quality of the individual probes.

3.4.1 Quality of the Overall Hybridization Experiment

We start by evaluating the distribution (or spreading) of the DNA fragments over the tiling array. This can be achieved by visual inspection of the array image within each separate channel (Fig. 2a).

By design, the signals should be randomly distributed and show no systematic spatial patterns. Artefacts such as scratches and bright spots can be easily detected in this way. The following commands are used to import the data in the R workspace:

```
>hybr_artefact<-read.table(file="hybridization_
artefact.txt",
+ header=TRUE,sep="\t")
> head(hybr_artefact)
```

```
     PROBE_ID            X      Y       PM
1    CHR01FS000000061    327    1335    3219.96
2    CHR01FS000000212    191    1257    4840.31
3    CHR01FS000000382    826    34      8668.02
4    CHR01FS000000507    731    529     19781.76
5    CHR01FS000000707    624    562     1195.29
6    CHR01FS000000827    927    485     7460.27
```

Plotting the reconstructed array image involves log transformation of the measured signals (PM) and rescaling of the log transformed signal between 0 and 1 in order to convert the signal into RGB colors. The code for plotting the array image (Fig. 2a) can be found at the above URL (*see* end of Subheading 2.3).

We also evaluate whether a sufficient amount of DNA was hybridized to the array. This can be done by plotting the density of the signal of each separate channel (Fig. 2b). The detection range of the signal has a lower and upper bound. In the case of insufficient DNA, there will be a rapid increase of probe signals in the lower detection range. Conversely, in the case of too much DNA the signal distribution will become saturated in the upper detection range. Both scenarios can seriously compromise the sensitivity of the technology to capture biologically meaningful variation. To illustrate this, we plot the density of the input signal of two different arrays (Fig. 2b) using the plotting code that is provided as a text file (*see* end of Subheading 2.3).

3.4.2 Quality of Individual Probes

The second quality assessment level is the quality of the probes. NimbleGen arrays are designed to minimize cross-hybridization as much as possible. However, given the large number of probes and near full genome coverage, it is difficult to exclude possible cross-hybridization events. Such events occur when nontarget sequences hybridize with probes on the array, leading to exaggerated signal intensities. It may therefore be desirable to identify probes, a priori, that have multiple similar or exact matches in the genome. We assess this by calculating the so-called conservation score. This score is obtained by performing a BLAST search. Scores are percentage of

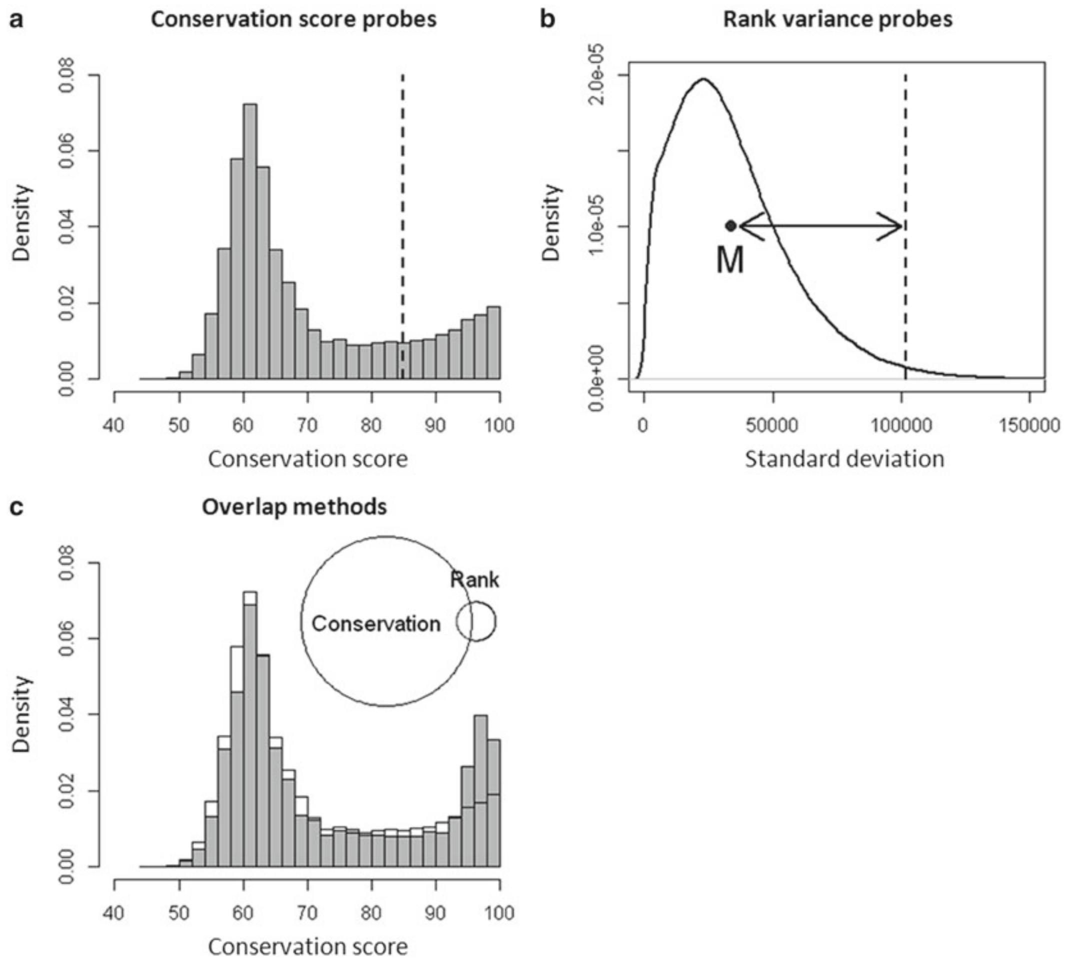

Fig. 3 Quality of individual probes. (**a**) Density histogram of the conservation score of the probes. Probes with a conservation score higher than 85 have a high probability to cross-hybridize and are flagged (probes on the *right* of the *dashed line*). (**b**) The rank variance distribution of the probes. The rank variance is expressed as a standard deviation. Probes with an abnormal high rank variance are flagged (probes on the *right* of the *dashed line*). (**c**) Density histogram of the conservation score of the rank variance probes that were flagged (*gray*) on *top* of the conservation score of all probes (*transparent*). This picture indicates that the probes with a high rank variance also tend to have a high conservation score. The Venn diagram shows however that there is a poor overlap between probes that are flagged with the two methods

identity with the second best hit (the best hit is the location on the genome for which the probe was designed). We decided to flag probes that have a conservation score higher than 85 (Fig. 3a). For simplicity, we here provide a complete dataset with conservation scores already assigned to each probe.

This data can be inputted as follows:

```
> cons_score_probes  <-  read.table(file="conservation_
  score.txt",
+ header=TRUE,sep="\t")
> head(cons_score_probes)
```

```
     PROBE_ID              SCORE
1    CHR01FS000000061      73
2    CHR01FS000000212      56
3    CHR01FS000000382      64
4    CHR01FS000000507      62
5    CHR01FS000000707      74
6    CHR01FS000000827      66
```

One can use the plotting code which is provided as a text file to plot the density histogram of the conservation scores of the probes as shown in Fig. 3a.

In addition to the above a priori screening of potential cross-hybridizing probes, we utilize another quality criterion, which involves assessing the consistency of probe signals for the INPUT across biological or technical replicates (provided they are available). To do this, we identify a probe's signal rank in the overall array signal distribution of one replicate array and compare it to its rank in the distribution of the other arrays. Inconsistent probe signals will show large variation in ranks and should be treated with caution. If we consider the three dye-swapped biological replicates (3 × 2 arrays) of the Col-0 accessions, there are six rank values for each probe, and we can calculate its rank variance. Doing this for each probe on the array yields a rank variance distribution, which can be used to spot outlying probes (Fig. 3b). For example, we may want to consider excluding or flagging probes with a rank variance of more than 3 standard deviations from the mean (Fig. 3b).

We use the following code to determine the rank and the rank variance of the probes as well as the three standard deviation cutoff:

```
> probe_rank         <-     apply(log2_input_wt[,2:7],
                     MARGIN=2,rank)
> determine_rank_var <- function(x){
+    mean_val       <- mean(x)
+    mean_dif       <- abs(x-mean_val)
+    extreme        <- which(mean_dif == max(mean_dif))
+    sd_ext         <- sd(x[-extreme])
+    return(sd_ext)
+ }
```

```
> rank_var          <-     apply(probe_rank,MARGIN=1,
                    determine_rank_var)
> rank_var          <-  data.frame(log2_input_wt[,1],
                    rank_var)
> names(rank_var)   <- c("PROBE_ID","SD")
> mean_var          <- mean(rank_var[,2])
> sd_var            <- sd(rank_var[,2])
> sd_cutoff         <- mean_var+(3*sd_var)
```

The plot of the rank variances (Fig. 3b) can be generated using the plotting code that is provided as a text file (*see* end of Subheading 2.3).

We find that the use of conservation scores and probe rank variance provides a fairly comprehensive assessment of probe quality. That these two criteria are not redundant is reflected in the limited overlap of identified low quality probes (Fig. 3c). We determine this overlap using the following code:

```
> lowq_pr_rank  <- rank_var[which(rank_var[,2]  > sd_
                 cutoff),1]
> lowq_pr_cons  <- cons_score_probes[which(cons_score_
                 probes[,2] > 85),1]
> lowq_probes   <- union(lowq_pr_rank,lowq_pr_cons)
> num_rank      <-  length(setdiff(lowq_pr_rank,lowq_
                 pr_cons)) #Only rank
> num_cons      <-  length(setdiff(lowq_pr_cons,lowq_
                 pr_rank)) #Only cons
> num_overlap   <- length(intersect(lowq_pr_rank,lowq_
                 pr_cons))
> lowq_rows     <-  which(wt_green_red[,1]  %in%  lowq_
                 probes)
```

Finally, we plot the overlap between the two methods (Fig. 3c) using the plotting code.

3.4.3 The Effect of Removing Low Quality Probes

The removal of low quality probes has a visible impact on the overall signal distribution. To see this, we plot the relative (or ratio) signal of the IP and the INPUT channel in Fig. 4 on a $\log_2$ scale (*see* file with plotting code). High signals are typically an indication of increased IP hybridization events relative to the total (INPUT) DNA, thus indexing methylated DNA sequences. We find that most low quality scores fall in the upper signal range, suggesting that true binding events are partially confounded with cross-hybridization events. This is consistent with the observation, in *Arabidopsis*, that DNA methylation primarily occurs in CG-rich repeat elements [3, 4], which have a high cross-hybridization potential. For all subsequent analysis, we decided to keep (but flag) low quality probes in the dataset. However, one may also choose to exclude them at this stage.

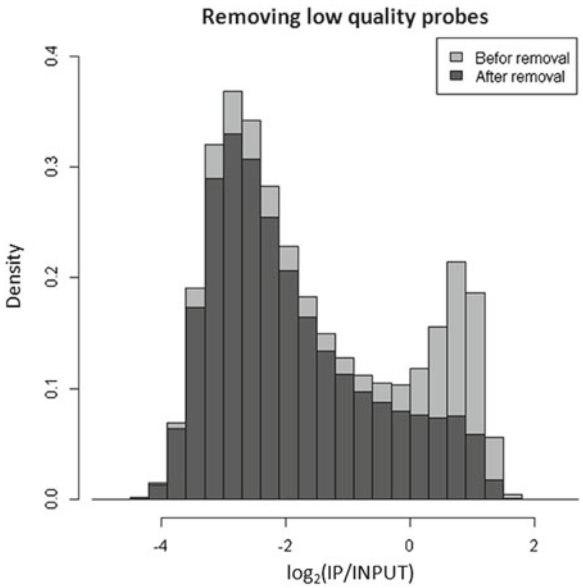

Fig. 4 Effect of removing low quality probe signals from the overall $\log_2$(IP/INPUT) signal distribution. Most low quality signals fall in the upper range of the distribution, suggesting that true binding events are partially confounded with cross-hybridization events

3.5 Implementation of a Hidden Markov Model for Reconstructing the DNA Methylome

The above-mentioned log2 transformed IP/INPUT signal ratio is the typical starting point for data analysis. If data from several replicates is available, as in our case, the probe signals can simply be averaged across replicates. We view this distribution (*see* Fig. 4) as a mixture of three partially overlapping components [6]. The right component corresponds to enriched probes (i.e., indexing methylated sequences), the left component to non-enriched probes (i.e., indexing unmethylated sequences), and the middle component to intermediately enriched probes (i.e., indexing intermediately methylated sequences). To illustrate that this mixture view is consistent with the underlying biology, we highlight the probe signals corresponding to annotated transposable elements, which are usually methylated in *Arabidopsis* (Fig. 5a, solid line; [4, 5]). Similarly, as an example of usually unmethylated sequences, we highlight the signal of annotated introns of protein-coding genes (Fig. 5a, dashed line; [4, 5]).

The following commands are used to import the probe annotation data. These files simply contain the probe identifiers of probes that match with introns or transposons.

```
> p_id_intron <- read.table(file="intron_probes.txt",
+ header=TRUE,sep="\t")
> p_id_transp <- read.table(file="transposon_probes.txt",
+ header=TRUE,sep="\t")
    > head(p_id_intron)
    PROBE_ID
1 CHR01FS000004351
2 CHR01FS000005311
```

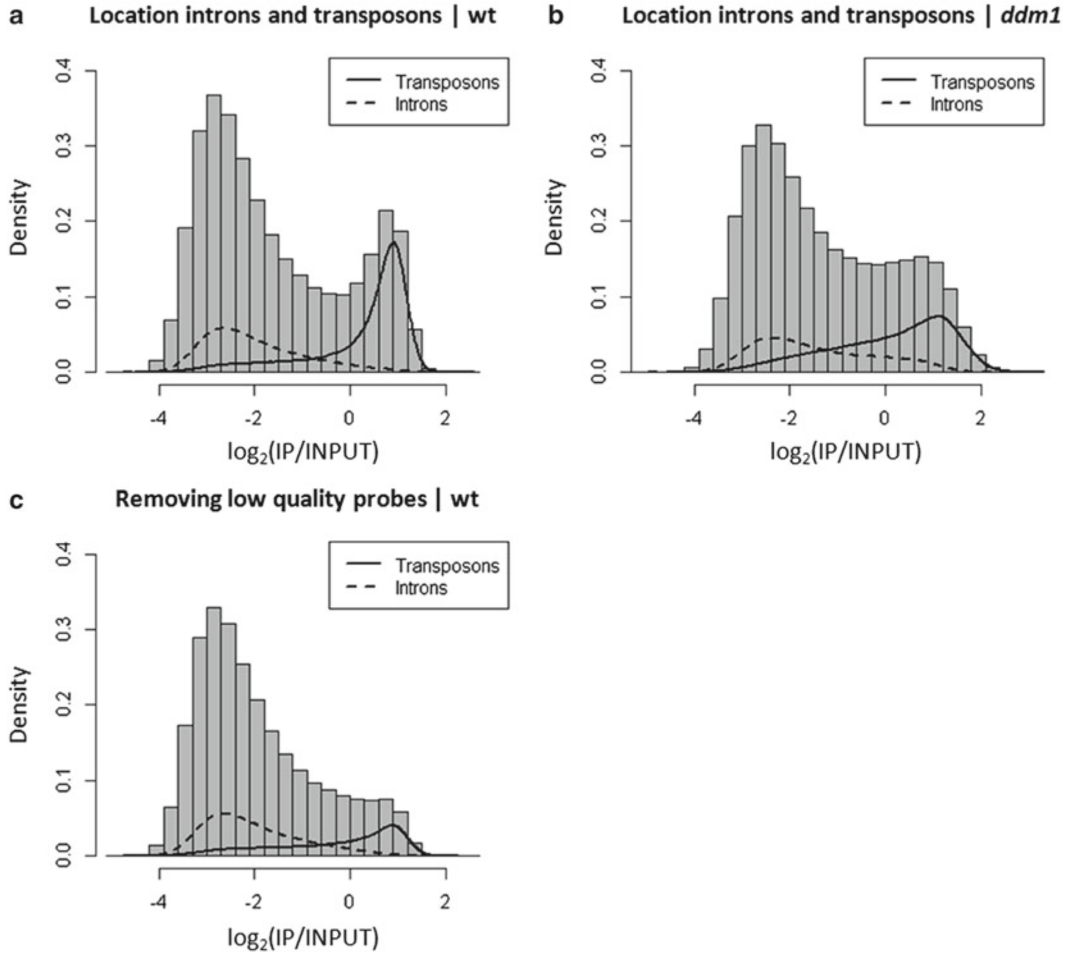

Fig. 5 Probe signal distributions of transposable elements and introns. (**a**) The $\log_2$(IP/INPUT) signal distribution of one dye combination (IP: *green*, INPUT: *red*) of the wild type Columbia plant with transposons (*solid*) and introns (*dashed*) highlighted. (**b**) Same as in (**a**) but shown for a *ddm1* mutant plant which has lost 70 % of its DNA methylation. The intron distribution is not much affected by this loss. (**c**) Same as in (**a**) but with low quality probe signals removed. As can be seen, the intron distribution is robust to low quality probe signals

```
3 CHR01FS000007129
4 CHR01FS000007479
5 CHR01FS000007814
6 CHR01FS000008139
```

For plotting purposes and further analysis steps, it is also necessary to know the rows of the probes that correspond to transposons or introns. The following commands determine those rows:

```
> rows_intron      <-  which(wt_green_red[,1]  %in%
                       p_id_intron[,1])
> rows_transp      <-  which(wt_green_red[,1]  %in%
                       p_id_transp[,1])
```

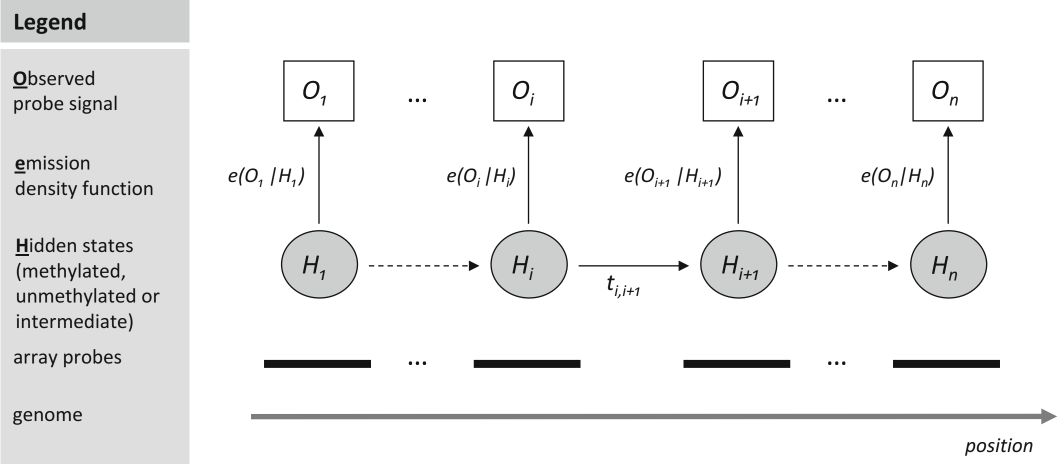

Fig. 6 Schematic of a HMM model in the context of genome-wide tiling array data. An explanation of the different components of the HMM is provided in the figure

```
> rows_intron_highq    <- setdiff(rows_intron,lowq_
                         rows)    #Without flagged
> rows_transp_highq    <- setdiff(rows_transp,lowq_
                         rows)    #probes
```

The file with plotting code contains the code for plotting Fig. 5.

As can be seen in this figure, even within these two extreme annotation sets (i.e., transposons and introns) there is substantial signal variation (Fig. 5). This is probably due to some level of biological variation (i.e., not all transposable element sequences are methylated and not all introns are unmethylated), but it certainly also reflects the limitations of the measurement technology itself [6]. In addition, many probe signals belong to annotation sets that cannot be easily assigned to these extreme mixture components, and their classification as methylated, unmethylated, or intermediate is inherently probabilistic.

Our principle analytical approach for performing this probabilistic classification relies on the use of a HMM. A Markov chain is a list of random values {H_1, H_2, …, H_n} that satisfy the so-called Markov property: the value at position i (H_i) is related solely to the values at positions $i-1$ and $i+1$ (H_i-1 and H_i+1), with given transition probabilities. In the case of a Hidden Markov chain, an output {O_1, O_2, …, O_n} is observed that depends on the unobserved (hidden) states of the chain, {H_1, H_2, …, H_n} [8]. In the case under consideration, the output or observed chain is the $\log_2$ transformed IP/INPUT signal ratio, while the hidden chain is the methylation state of the DNA sequence indexed by the array probe (Fig. 6).

Hence, the HMM approach capitalizes on two key properties of MeDIP-chip data: (1) probe signals are noisy proxies of an unobserved (hidden) methylated, intermediate, or unmethylated state, and (2) the probe signals are spatially correlated along the genome, so that neighboring probes provide similar information (Fig. 6). HMMs account for these two properties and provide a powerful statistical framework for classifying individual probe signals given the overall data structure. Our implementation goal is to provide a robust and fast model estimation procedure. We achieve this by implementing software code in C++ and by incorporating several useful biological constraints. In what follows, we outline our version of a HMM that is specifically designed for *Arabidopsis* NimbleGen MeDIP-chip data. We start by detailing key data preparation steps before we move on to discuss the actual implementation strategy.

3.5.1 Data Rescaling Using Intron Probes

In the context of a single MeDIP experiment within-array normalization is not required in our experience. Nonetheless, we find that rescaling the overall signal distribution is generally a good idea to permit more meaningful comparisons across different individuals (i.e., experimental conditions), should such additional data become available. To achieve this, we make use of the intron probe signal distribution (Fig. 7a). We standardize this distribution and express the overall signal distribution in terms of their standard deviation values. This has the effect of placing the mean of the intron probe signal at zero and rescaling the values as standard deviation values. This rescaling can be implemented with the following code:

```
> intron_mean   <- mean(wt_dye_swap[rows_intron,2])
> intron_sd     <- sd(wt_dye_swap[rows_intron,2])
> wt_dye_swap_rs <-     (wt_dye_swap[,2]-intron_mean)/
                  intron_sd
```

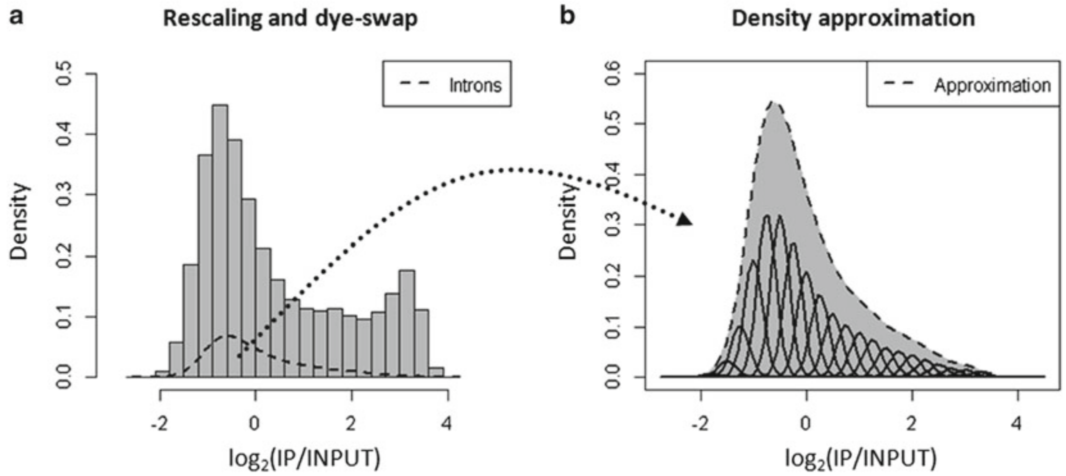

Fig. 7 Data rescaling and density approximation probe signal distribution of introns. (**a**) Original signal distribution with intron density highlighted (*dashed line*). (**b**) Density of the signal distribution for introns approximated using a mixture of a large number of Gaussian distributions with fixed variance and equally spaced means

```
> wt_dye_swap_rs        <- data.frame(wt_dye_swap[,1],
                           wt_dye_swap_rs)
> names(wt_dye_swap_rs)  <- c("PROBE_ID","DYE_SWAP_RS")
```

One can use the plotting code that is provided as a text file to plot the density of the rescaled data (Fig. 7a).

We find that the intron signal distribution can be safely used for this rescaling process, insofar that it is relatively invariant to high levels of experimental variation. To illustrate this in the context of an extreme case, we compare the signal distribution for wild type to that for the *ddm1* mutant, in which DNA methylation is reduced approximately 70 %. The MeDIP-chip experiment reflects this methylation loss nicely (Fig. 5b), with the signal distribution being clearly reduced in height over the enriched component. Clearly, the signal distribution for intronic sequences is not noticeably affected in *ddm1*, as expected.

3.5.2 Implementation of the Hidden Markov Model

We apply our HMM to the rescaled data following a two-step process. First, we use the Baum–Welch algorithm [8, 9] to estimate the best model parameters given the observed probe signals (Fig. 6). Second, we find the most likely hidden sequence of probe states given these estimated parameters. A copy of the C++ code that we implement ccan be found at the above URL (*see* end of Subheading 2.3).

A characteristic feature of our HMM implementation is the use of biologically meaningful constraints on the emission probability density functions, e(O_i |H_i), during the Baum–Welch estimation procedure (Fig. 6). In the following we outline these assumptions. A summary of them can be found in Table 5. Alternatively, all the parameters of the emission probabilities could be freely estimated by means of the Baum–Welch algorithm, but we find that a more biologically meaningful approach is preferable.

Emission probability of unmethylated hidden state: We employ the signal distribution for introns to obtain an approximation of

Table 5
Summary of the constraints for the emission probability density functions used in the Baum–Welch algorithm

Hidden state	Distribution	Parameters
Unmethylated states	Intron signal distribution	Estimated as a mixture of 30 normals (EM algorithm) with fixed variance
Methylated states	Gaussian	Mean: fixed at the 99th quantile of the intron signal distribution. Variance: freely estimated
Intermediate states	Gaussian	Mean: fixed at ½ (mean of the methylated distribution). Variance: equal to the variance of the methylated distribution

the emission probability of the unmethylated hidden state (Fig. 7a). In this way we incorporate biological knowledge of introns being mostly unmethylated directly into the estimation procedure. This bypasses the need to explicitly assume an emission density function, and also speeds up computation. We approximate the signal distribution for introns to an arbitrary degree using mixtures of a large number of Gaussian random variables (Fig. 7b). Estimation is carried with the EM algorithm [10], which can be implemented using the following code:

```
> density_approx   <- function(data,mu,var,lambda,eps,
                        num_norm){
+   mu_diff  <- mu[2]-mu[1]
+   min_val  <- mu[1]-5*mu_diff
+   max_val  <- mu[num_norm]+5*mu_diff
+   rows_extr       <- which(data < min_val | data >
                        max_val)
+   if(length(rows_extr) > 0){   # Remove extreme
+    data    <- data[-rows_extr] # data points
+   }
+   loglik_diff     <- 100000     # Initial loglik diff
+   counter      <- 0       # Iteration counter
+   dnorm_tot       <- rep(0,length(data))
+   for(A in 1:num_norm){
+     dnorm_tot      <-
+     dnorm_tot + lambda[A]*dnorm(data,mean=mu[A],sd
      =sqrt(var))
+   }
+   loglik_pre      <- sum(log(dnorm_tot))    #
    Initial loglik
+   while(loglik_diff > eps){   # Estimate mixture
+     counter       <- counter+1
+     for(A in 1:num_norm){     # Update lambda
+       post        <-          # Posterior prob
+       lambda[A]*dnorm(data,mean=mu[A],sd=sqrt(
        var))/dnorm_tot
+       lambda_new  <- sum(post)/length(data)
+       lambda[A]  <- lambda_new
+     }
+     dnorm_tot      <- rep(0,length(data))
+     for(A in 1:num_norm){
+       dnorm_tot    <-
+       dnorm_tot + lambda[A]*dnorm(data,mean=mu[A],
        sd=sqrt(var))
+     }
+     loglik_new    <- sum(log(dnorm_tot))          #
      New loglik
+     loglik_diff    <- abs(loglik_new - loglik_pre)
      # New loglik diff
+      loglik_pre    <- loglik_new
+      cat("Iteration  =  ",counter,"  Log-lik  diff  =
       ",loglik_diff,"\n")
+     }
+    output   <- list(mu,var,lambda)       # Return results
+     names(output) <- c("mu","var","lambda")
+     return(output)
+ }
 >
> mus <- round(seq(-2.75,4.5,0.25),2)
> intron_data <- wt_dye_swap_rs[rows_intron,2]
> den_appr    <- density_approx(data=intron_data,mu=mus,
              var=0.03,
+
lambda=rep((1/30),30),eps=0.1,num_norm=30)
Iteration =  1  Log-lik diff =  64314.34
Iteration =  2  Log-lik diff =  510.154
Iteration =  3  Log-lik diff =  41.84451
Iteration =  4  Log-lik diff =  6.50513
Iteration =  5  Log-lik diff =  1.766472
Iteration =  6  Log-lik diff =  0.7541914
Iteration =  7  Log-lik diff =  0.421424
Iteration =  8  Log-lik diff =  0.2723324
Iteration =  9  Log-lik diff =  0.1922282
Iteration =  10  Log-lik diff =  0.1442565
Iteration =  11  Log-lik diff =  0.1132907
Iteration =  12  Log-lik diff =  0.09209521
 >
```

The code for plotting the result (Fig. 7b) is provided as a text file.

We generally find that a fit with 30 Gaussians with fixed variance provides a sufficient approximation (Fig. 7b). Parameter estimates are outputted to be used as input in the Baum–Welch algorithm.

Emission probability of methylated hidden state: The second constraint relates to the emission probability for the methylated hidden state. We assume this distribution to be Gaussian, with mean fixed to the 99th quantile of the emission probability of the unmethylated state (i.e., the signal distribution for introns). The variance of the distribution is estimated freely by the Baum–Welch algorithm.

Emission probability of intermediate hidden state: The last constraint relates to the emission probability for the intermediate hidden state. We assume again this distribution to be Gaussian, with a mean that is fixed between the mean of the emission probability of the unmethylated hidden state (i.e., the intron distribution) and the mean of the emission probability of the methylated hidden state. We take the variance of this distribution to be equal to the variance of the emission probability of the methylated hidden state.

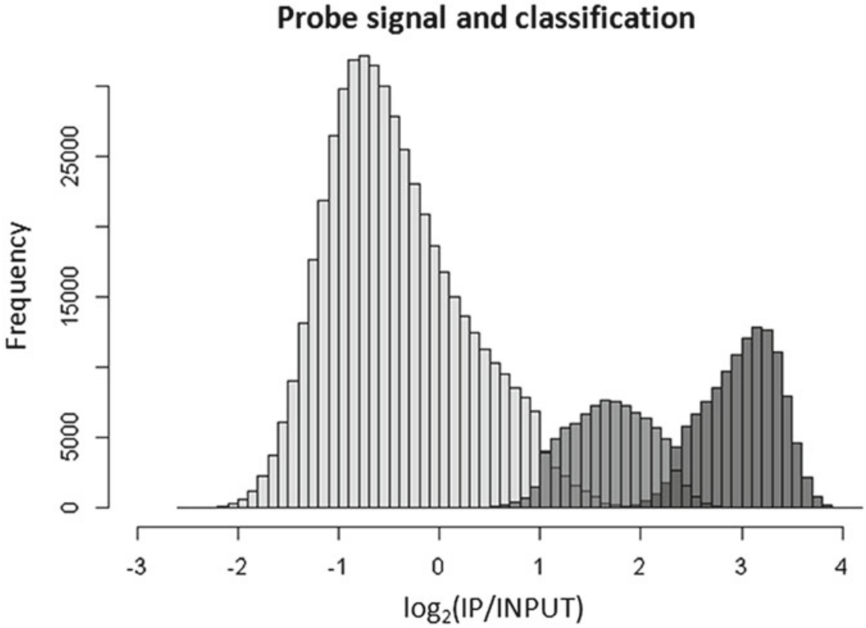

Fig. 8 The $\log_2$(IP/INPUT) signal distribution for a wild type *A. thaliana* Col-0 accession. Probes are classified into unmethylated probes (*light gray*), intermediate probes (*gray*), and methylated probes (*dark gray*)

The following code generates files that are used as input for the Hidden Markov program written in C++:

```
> values <-          c(den_appr$mu,den_appr$var,den_
           appr$lambda)
> parameters <- c(paste("mu",1:30,sep=""),"var_all",
+ paste("lambda",1:30,sep=""))
> para_est <- data.frame(parameters,values)
> names(para_est) <- c("PARAMETER","VALUE")
> write.table(para_est,"para_wild_type.txt",quote=FALSE,
  sep="\t",
+ row.names=FALSE,col.names=TRUE)
> write.table(wt_dye_swap_rs,"dye_swap_signal_wild_
  type.txt",
+ quote=FALSE,sep="\t",row.names=FALSE,col.
  names=TRUE)
```

Once all the free parameters of the HMM have been estimated, we proceed to infer the most likely hidden sequence of probe states given the parameters of the HMM and the observed probe signals. There are several possible strategies, depending on our optimality criterion. We consider two cases: (1) finding the single best hidden sequence of probe states, given the observed probe signals and the parameters of the HMM (the so-called Viterbi algorithm) (Viterbi 1967, Rabiner 1989), or (2) finding the single hidden probe state which is individually most likely at each position, given the observed probe signals and the parameters of the HMM (Rabiner 1989). A copy of the C++ code implemented for the

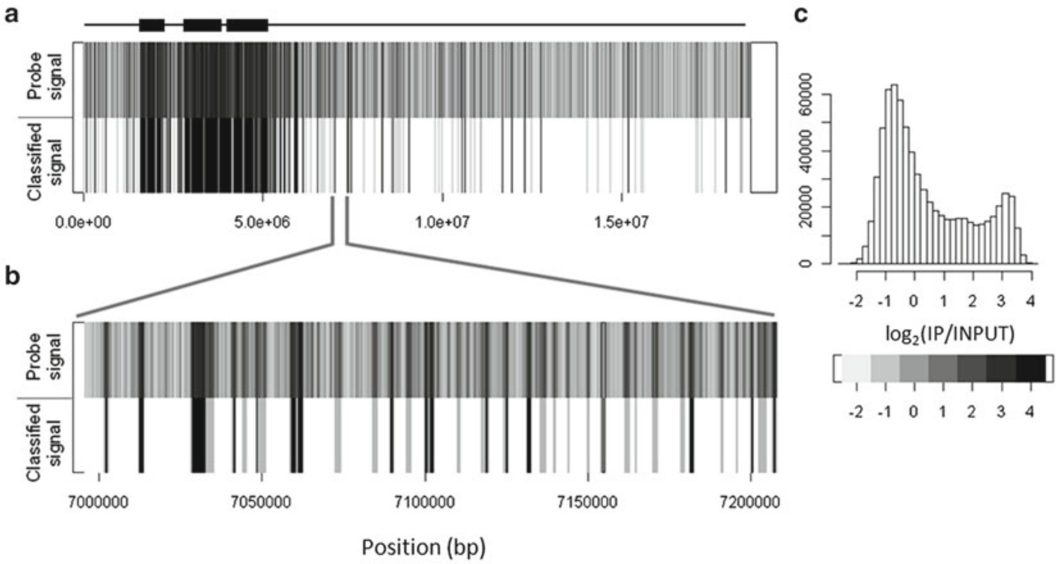

Fig. 9 Example probe classification. (**a**) The probe signal (*top*) and the corresponding (*hidden*) DNA methylation state (*bottom*) of chromosome 4 in wild type Columbia accession plotted against position (base pairs, *x*-axis); methylated (*black*), unmethylated (*white*), and intermediately methylated (*gray*). As expected, we find substantial methylation in the pericentromeric regions as well as in the heterochromatic knob present on the short arm of the chromosome. (**b**) Magnification of a small region on chromosome 4: we can see how the $\log_2$(IP/INPUT) signal of each probe (*top*) is assigned to methylated, intermediate, or unmethylated state, depending on its signal and on the signal of its surrounding probes. (**c**) Color code for the probe signal density plot, with the corresponding probe density distribution

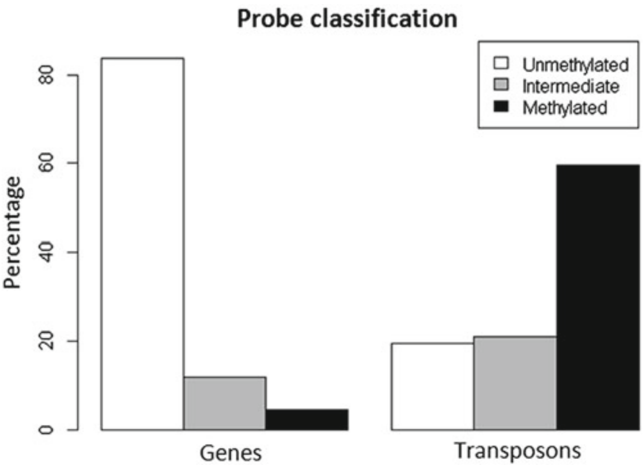

Fig. 10 Probe classification of genes and transposable elements

identification of the optimal sequence according to these two definitions can be found at the above URL (*see* end of Subheading 2.3).

3.6 Graphical and Biological Assessment of HMM Results

The above algorithms probabilistically classify the original log2(IP/INPUT) signals to the three underlying methylation states (unmethylated, intermediate, or methylated) (Fig. 8). This "hidden chain" of methylation states constitutes the methylome (Fig. 9). Annotation analysis of the probe classification (Fig. 10) shows that most gene probes are unmethylated and the majority of the transposable element probes are methylated, as expected.

4 Conclusions

We have described a comprehensive protocol for the analysis of DNA methylomes in *Arabidopsis* using MeDIP tiling arrays. Our protocol uniquely combines all necessary steps from "wet lab" to "dry lab" to begin to characterize the epigenetic landscape in this species. Owing to the relatively favorable cost of tiling array technology over more recent deep sequencing approaches, our protocol can be easily scaled up to population-level studies. Such large epigenetically informative approaches will soon become an indispensable tool in the context of intra- or intergenerational functional studies [12]. We have applied the protocol outlined here to a large panel of epiRILs [3] in order to characterize the role of DNA methylation in complex trait inheritance.

5 Notes

1. For more than 250 μL of beads, separate in two tubes for washes.
2. Transfer to new tubes decreases noise. This is done in classical tubes because siliconized tubes tend to leak too much with phenol/chloroform and can cause loss of material.
3. MinElute cleaning is a critical step as the efficiency of WGA2 drops dramatically without it.
4. PicoGreen quantification is very sensitive. Be careful to homogenize your samples well before quantification. Since PicoGreen is not stable in light, quantification must be done soon (less than 30 min) after addition of PicoGreen and samples should be maintained in the dark before use.
5. It is important to verify incorporation of dye using the following formula: concentration in DNA (pmol/μL)/concentration in Dye (pmol/μL). Values are usually between 100 and 180.

Acknowledgements

This work was supported in part by grants from the Agence Nationale de la Recherche (Genoplante TAG project, to V.C.) and by the European Union Network of Excellence "The Epigenome" (to V.C.). S.C. was supported by a Ph.D. studentship from the Ministère de l'Enseignement Supérieur et de la Recherche. R.W., M.C.-T., and F.J. were supported by grants from The Netherlands Organisation for Scientific Research.

References

1. Johannes F, Porcher E, Teixeira FK, Saliba-Colombani V, Simon M, Agier N, Bulski A, Albuisson J, Heredia F, Audigier P, Bouchez D, Dillmann C, Guerche P, Hospital F, Colot V (2009) Assessing the impact of transgenerational epigenetic variation on complex traits. PLoS Genet 5:e1000530
2. Reinders J, Wulff BB, Mirouze M, Marí-Ordóñez A, Dapp M, Rozhon W, Bucher E, Theiler G, Paszkowski J (2009) Compromised stability of DNA methylation and transposon immobilization in mosaic *Arabidopsis* epigenomes. Genes Dev 23:939–950
3. Colomé-Tatché M, Cortijo S, Wardenaar R, Morgado L, Lahouze B, Etcheverry M, Martin A, Feng S, Duvernois-Berthet E, Labadie K, Wincker P, Jacobsen SE, Jansen RC, Colot V, Johannes F (2012) Features of the Arabidopsis recombination landscape resulting from the combined loss of sequence variation and DNA methylation. Proc Natl Acad Sci USA 109:16240–16245.
4. Cokus SJ, Feng S, Zhang X, Chen Z, Merriman B, Haudenschild CD, Pradhan S, Nelson SF, Pellegrini M, Jacobsen SE (2008) Shotgun bisulphite sequencing of the *Arabidopsis* genome reveals DNA methylation patterning. Nature 452:215–219
5. Lister R, O'Malley RC, Tonti-Filippini J, Gregory BD, Berry CC, Millar AH, Ecker JR (2008) Highly integrated single-base resolution maps of the epigenome in *Arabidopsis.* Cell 133:523–536
6. Johannes F, Wardenaar R, Colomé-Tatché M, Mousson F, de Graaf P, Mokry M, Guryev V, Timmers HT, Cuppen E, Jansen RC (2010) Comparing genome-wide chromatin profiles using ChIP-chip or ChIP-seq. Bioinformatics 26:1000–1006
7. Laird PW (2010) Principles and challenges of genome-wide DNA methylation analysis. Nat Rev Genet 11:191–203
8. Rabiner LR (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. Proc IEEE 77: 257–286
9. Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41: 164–171
10. McLachlan GJ, Peel D (2000) Finite mixture models. John Wiley and Sons, Inc.
11. Viterbi AJ (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans Inf Theory 13:260–269
12. Johannes F, Colot V, Jansen RC (2008) Epigenome dynamics: a quantitative genetics perspective. Nat Rev Genet 9:883–890