extract and utilize probe-level information across large microarray data collections in a fully scalable manner.

To overcome the key limitations of the current approaches, we introduce a fully scalable algorithm for multi-array preprocessing based on Bayesian online-learning. The new algorithm extends the Robust Probabilistic Averaging (RPA) framework introduced in (21) by providing a model for probe affinities and by incorporating prior terms to provide the basis for scalable online-learning through sequential hyperparameter updates. The resulting algorithm allows rigorous preprocessing of very large microarray atlases on an ordinary desktop computer in small consecutive batches with minimal memory requirements and in linear time with respect to sample size. In contrast to the currently available alternatives, the proposed Online-RPA algorithm provides the means to integrate probe-level information across large-scale microarray atlases involving tens or hundreds of thousands of arrays and it is readily applicable to all short oligonucleotide microarray platforms. In addition, the analysis of probe performance can now be based on the most comprehensive collections of microarray data to guide microarray design and quality control. To our knowledge this is the only probe-level preprocessing method which is both fully scalable and platform-independent, providing new tools to take full advantage of contemporary microarray data collections.

MATERIALS AND METHODS

Probe-level procedures that combine information across multiple arrays have been found to improve preprocessing performance (13) but their applicability to large sample collections has been limited due to huge memory requirements associated with increasing sample sizes. The available solutions have been based on learning and extrapolation of probe-level effects from smaller reference training sets (18, 19). In this section, we outline an alternative online-learning procedure that can extract and utilize probe-level information across very large microarray collections in a fully scalable manner with minimal memory requirements and in linear time with respect to sample size based on Bayesian hyperparameter updates.

Scalable preprocessing with online-learning: an overview

In the following, let us outline the proposed online-learning procedure and provide details of parameter estimation. Assuming that appropriate microarray quality controls have been applied prior to the analysis, the standard steps of background correction, normalization, and probe summarization are applied to consecutive batches of the data in three sweeps over the data collection:

Step 1: Background correction and quantile basis estimation In the first step, each individual array is background-corrected. In the present work we use the standard RMA background correction (14). The background corrected data is stored temporarily on hard disk to speed up preprocessing. The basis for quantile normalization is then obtained by averaging sorted probe-level signals from background-corrected data (13). For scalable estimation of the base distribution, we average over the estimates from individual batches to obtain the final quantile base distribution as in parallel implementations of RMA (22). The final base distribution is identical with the one which would be obtained by jointly normalizing all arrays in a single batch.

Step 2: Hyperparameter estimation The key novelty of our approach is in introducing the scalable approach for estimating the probe-level hyperparameters. This is achieved based on Bayesian online-learning where consecutive batches of data are used to update the hyperparameters of the model. Before hyperparameter estimation, each batch is background-corrected, quantile-normalized and log-transformed. At the first batch, the model can be initialized by giving equal priors for the probes if no probe-specific prior information is available. The probe-level hyperparameters are then updated at each new batch, and provided as priors for the next batch. The final probe-level parameters are obtained after processing the complete data collection. Ideally, the fully scalable parameter estimation through consecutive hyperparameter updates will yield identical results with a single-batch approach.

Step 3: Probe summarization The final probe-level parameters from the second step are used to summarize the probes in each batch, yielding the final preprocessed data matrix.

The probe-level model

Let us first summarize the probe-level model for a fixed probeset with J probes across $T+1$ arrays. The model assumes background-corrected, normalized, and log-transformed probe-level data. The algorithm is based on a Gaussian model for probe effects, where the signal s_{ij} of probe $j \in \{1,\dots,J\}$ in sample $i \in \{1,\dots,T+1\}$ is modeled as a sum of the underlying target signal a_i and Gaussian mean and variance parameters μ_j, τ_j^2 that are directly interpretable as constant affinity μ_j and stochastic noise $\varepsilon_{ij} \sim N(0,\tau_j^2)$, respectively:

$$s_{ij} = a_i + \mu_j + \varepsilon_{ij} \sim N(a_i + \mu_j, \tau_j^2). \qquad \mathbf{(1)}$$

In the following, let us outline the estimation procedure for the model parameters $\mathbf{a} = [a_1,\dots,a_{T+1}]$, $\boldsymbol{\mu} = [\mu_1,\dots,\mu_J]$, $\boldsymbol{\tau}^2 = [\tau_1^2,\dots,\tau_J^2]$. We start by estimating the variance parameters $\boldsymbol{\tau}^2$ of this model by following (21), and additionally incorporate Bayesian prior terms in the model to obtain a fully scalable algorithm. Affinity estimation ($\boldsymbol{\mu}$) relies on the final probe-specific variance estimates. The final probeset-level summaries are obtained after estimating the probe-specific affinity and variance parameters.

Incorporating prior information of the probes Estimation of the probe-specific variance parameters is based on probe-level differential expression signal $s_{tj} - s_{rj}$ between each sample $t = [1,\dots,T]$ and a randomly selected reference sample r. Then, given Eq. **1**, the unidentifiable affinity parameters μ_j cancel out, yielding $s_{tj} - s_{rj} = (a_t - a_r) + (\varepsilon_{tj} - \varepsilon_{rj})$. Following (21), let us denote $m_t = s_t - s_r$, $d_t = a_t - a_r$, and apply the vector notation $\mathbf{m} = [m_1,\dots,m_T]$, $\mathbf{d} = [d_1,\dots,d_T]$. Then the full posterior density for the model parameters $\mathbf{d}, \boldsymbol{\tau}^2$ is obtained with the Bayes' rule as