SureTypeSC - A Random Forest and Gaussian Mixture predictor of high confidence genotypes in single cell data

Ivan Vogel[1,2] , Robert C. Blanshard[3,4] and Eva R. Hoffmann[1,4 *]

[1]DNRF Center for Chromosome Stability, Department of Cellular and Molecular Medicine, Faculty of Health and Medical Sciences, University of Copenhagen, 2200 Copenhagen, Denmark,

[2]Faculty of Information Technology, Brno University of Technology, Bozetechova 1/2, 616 66 Brno, Czech Republic,

[3]Illumina UK Ltd., Capital Park, Fulbourn, Cambridge, CB21 5XE, UK,

[4]Genome Damage and Stability Centre, School of Life Sciences, University of Sussex, Brighton, BN1 9RQ, UK.

*To whom correspondence should be addressed.

Abstract

Motivation: Accurate genotyping of DNA from a single cell is required for applications such as *de novo* mutation detection and linkage analysis. However, achieving high precision genotyping in the single cell environment is challenging due to the errors caused by whole genome amplification. Two factors make genotyping from single cells using single nucleotide polymorphism (SNP) arrays challenging. The lack of a comprehensive single cell dataset with a reference genotype and the absence of genotyping tools specifically designed to detect noise from the whole genome amplification step. Algorithms designed for bulk DNA genotyping cause significant data loss when used for single cell applications.

Results: In this study, we have created a resource of 28.7 million SNPs, typed at high confidence from whole genome amplified DNA from single cells using the Illumina SNP bead array technology. The resource is generated from 104 single cells from two cell lines that are available from the Coriell repository. We used mother-father-proband (trio) information from multiple technical replicates of bulk DNA to establish a high quality reference genotype for the two cell lines on the SNP array. This enabled us to develop SureTypeSC - a two-stage machine learning algorithm that filters a substantial part of the noise, thereby retaining of the majority of the high quality SNPs. SureTypeSC also provides a simple statistical output to show the confidence of a particular single cell genotype using Bayesian statistics.

Contact: eva@sund.ku.dk

1 Introduction

Single cell genomics is an umbrella term for genotyping of individual cells from a heterogeneous population. The deconvolution of mixed populations allows detection of genetic diversity within a population of cells. Applications cover many disciplines from sequencing the complete genomes of microorganisms that are challenging to culture in the laboratory to *de novo* mutation detection in tumour cells isolated from circulating blood. Detecting genomic changes in single cells is a sensitive procedure, complicated by the often rare, unique and precious nature of the starting material, such as genetic testing of human embryos for diagnostic purposes.

Unlike sequencing of bulk DNA, single cell sequencing requires a whole genome amplification (WGA) step to generate sufficient material for genotyping by next-generation sequencing (NGS) or single-nucleotide polymorphism (SNP) array (Gawad et al., 2016). A typical human cell contains 8 pg nuclear DNA that must be amplified to meet the input requirements for PCR- free sequencing (1 µg) or SNP array analysis (400 ng). The efficacy of genotyping from a single cell is critically dependent on the WGA method. Genome coverage, replication fidelity and the level of technical noise, such as systematic or stochastic amplification bias, are the main features considered when choosing the WGA method. However, all WGA methods deteriorate the signal from single cell. The signal deterioration potentially carries two risks: (a) sub-optimally amplified signal can lead to a complete loss of information about a particular locus, (b) uneven signal amplification of two alleles in a heterozygous locus can lead to an erroneous homozygous genotype call. The latter is called allele drop out and its incidence is up to 30 % of all typed SNPs from a single cell (Blanshard et al., 2018). SNP array technology allows the analysis of a wide range of genetic variants with good coverage in a fast and cost-efficient manner. There is a plethora of

tools and algorithms currently available for genotyping bulk DNA (Ritchie et al., 2011; Li et al., 2012). These algorithms are optimized for SNPs on the array and perform very well in terms of both call rates and sensitivity. However, an algorithm that is specifically designed for single-cell variant calling is currently missing. This is important because it is unclear how well the genotyping platforms deal with the biases introduced by whole-genome amplification of DNA from single cells. One solution is to include only SNP calls that are similar in properties to those from bulk DNA. This causes a substantive loss of data, however. It is also unclear how accurate genotyping is after the whole-genome amplification.

Genotyping from SNP arrays relies on detection of emission intensities (X and Y). Thus, when X and Y are equally intense and above a certain threshold, the genotype is inferred as heterozygous (AB). In contrast, when only X or Y is detected above a certain threshold homozygous genotypes are assigned (AA or BB). Current genotyping algorithms are based on two distinct approaches. Model-based algorithms do not require a training data set and assume that every SNP can be modeled from a linear combination of multiple multivariate components (Teo et al., 2007, Giannoulatou et al., 2008). Reference-based algorithms work under presence of a comprehensive reference database prior to genotyping. Parameters of these algorithms are inferred from a training dataset (HapMap) and are used for normalization of the raw data (Ritchie et al., 2009) or confidence measure of the genotype (Kermani, 2008). The training of the parameters can be performed via supervised machine learning methods, in particular neural networks (Kermani, 2008).

Here, we present a comprehensive database of 104 single cell samples from two different cell lines that we SNP-typed and compared with their reference genotype. This allowed us to create a database with two classes of calls: (a) high quality single cell calls and (b) misclassified single cell calls caused by deteriorated signal. We used both classes to develop a two layered algorithm that combines supervised machine learning method with model-based algorithm that is able to identify the noise in the single cell data coming from erroneous whole genome amplification and then assign a probability score of a SNP being correctly genotyped.

2 Materials and Methods

2.1 Cell lines and molecular methods

We generated genotypes from whole genome amplified DNA (from single cells) or genomic DNA from bulk extraction using the Infinium Karyomapping Assay Kit (Illumina Inc., California, US). We obtained EBV-lymphoblastoid cell lines GM07228 and GM12878 from the NIGMS Human Genetic Cell Repository at the Coriell Institute for Medical Research, New Jersey, USA, and cultured these according to the supplier's recommendation. All of the molecular methods and genotyping using GenCall for obtaining the SNP genotypes provided in the Supplemental information. Generation of high quality reference genotypes are also described in the Supplemental information.

2.2 MA transformation

The MA transformation is an application of the Bland-Altman transformation (Bland and Altman, 1999) that has been used extensively in the analyses of gene expression data when intensity values for two channels are compared using microarrays (red and green, referred to as X and Y, respectively).

Formally, we apply a linear-log transformation for every SNP i carrying a tuple of intensities (r_i, g_i) by calculating the values m_i and a_i, as follows:

$$m_i = \log_2(r_i) - \log_2(g_i)$$

$$a_i = \frac{1}{2}[\log_2(r_i) + \log_2(g_i)]$$

The m-feature has powerful discriminative ability to separate the three genotype clusters and is able to reduce variability between experiments and SNPs (Carvalho et al., 2007). The a-feature is a good general indicator of the signal quality (Ritchie et al., 2011).

2.3 Bioinformatics workflow for the machine learning algorithm

We developed a bioinformatics workflow with a supervised machine learning core that filters out the noise from the single cell data. The reference training intensities as well as validation intensities are first extracted from the intensity data (*.idat) files and subsequently genotyped using the GenCall algorithm implemented in GenomeStudio.The training data is then exported from GenomeStudio, transformed using MA transformation and fitted to a two-layered machine learning model. The results are subsequently tested on a set of independent single cell samples. The details of the workflow are shown in Figure S1.

2.4 Training and validation datasets

We created a reference genotype for both single cell lines (GM07228 and GM12878) using parental information and multiple technical replicates from bulk DNA (Supplemental Methods). We subsequently compared the reference genotype to our single cell datasets. More specifically, to every candidate single cell call for SNP *i* and sample *s* we assigned a label: $l_{i,s} \in \{True, False\}$, depending on the match or mismatch with the corresponding reference genotype call. The training dataset is then a triplet $(m_{i,s}, a_{i,s}, l_{i,s})$, whereas $(m_{i,s}, a_{i,s})$ are input features and $l_{i,s}$ is the output feature. We included all single cell calls with GC score above 0.01 totalling in 14,805,232 SNPs for training (GM07228) and 11,799,864 SNPs for validation (GM12878). Lowering the cut-off (and therefore including potentially poorly amplified SNPs) allowed us to capture the full error pattern. Table 1 gives a detailed overview of the used datasets. At this stage of the workflow (process named "Building training dataset" in the workflow in Figure S1) we were able to observe the error pattern in the single cell data and display it in the form of contour plots (Figure 1B,C). Note that we omit sample index *s* in further explanation, as we do not distinguish between the origins of SNPs from the training data set. Throughout the text, we always use notation $x \in X$ meaning that set of values is always written uppercase and element of the set is noted lowercase.

2.5 Supervised training using Random Forest

Random Forest is an ensemble supervised training method that is built from the collection (forest) of classification (decision) trees (Breiman, 2001). Each tree is trained on a different random subset of data and different subsets of input features. Although the training data only contain two input features (m and a) the preliminary analysis (Figure 1D) suggests that the function that separates the erroneous clusters (red areas) from the correct calls (blue areas) is non-linear. Random Forest (RF) has the ability to fit different trees to different parts of the input space and therefore approximate a non-linear separating function resulting in

increased classification accuracy. We used the implementation of Random Forest from the scikit package (Pedregosa et al., 2011) for fitting the training data. We adjusted the following parameters of the algorithm:

- the number of trees was set from 10 to 30; according to Oshiro et al., 2012 a theoretical upper limit is 128 trees and further increase in number of trees does not contribute to higher accuracy; however our data suggest that forests with more than 30 trees contribute minimally to the accuracy of the model but increase the size of the model substantially (data not shown)
- the number of features to consider when looking for the best split of the tree: 2

The prediction was evaluated in two ways - by stratified 10-fold cross-validation and with an independent single cell dataset. As the training data is highly unbalanced (despite the bias that is omnipresent, there are always more correct calls than miscalls), stratification in the cross-validation assures that there are same proportions of the correct calls and miscalls between the folds.

We used the following metrics for validation:

- Recall (or sensitivity): $\frac{TP}{TP+FN}$
- Precision: $\frac{TP}{TP+FP}$
- Specificity: $\frac{TN}{TN+FP}$
- F1 score as the harmonic mean of precision and recall
- Receiver Operating Characteristic curve (ROC), which shows sensitivity (true positive rate) as a function of 1-specificity (false positive rate)
- ROC-AUC score – area under the ROC
- Precision-Recall (PR) curve displays recall as function of precision
- Posterior Probability Matrix is a custom-defined metric showing the posterior probabilities $P(g_{ref}|g_{sc})$, where g_{ref} is the reference call (in rows) and g_{sc} is the single cell call (in columns); posterior probability in this context is a confidence measure of genotype g_{sc} having a truth value of g_{ref}
- Allele drop-in (ADI) is an erroneous change from homozygous to heterozygous genotype and can by calculated from the Posterior Probability Matrix by applying: $P(AA|AB) + P(BB|AB)$
- Allele drop-out (ADO) is an erroneous change from heterozygous to homozygous genotype and can be calculated from the Posterior Probability Matrix by calculating $P(AB|AA)$. $P(\mathrm{AA|hom}) + \mathrm{P(AB|BB)}$. $\mathrm{P(BB|hom)}$

TP, FN, TN and FP mean true positive, false negative, true negative and false positive respectively. The points of ROC and PR curve were drawn by applying various cut-offs of the algorithm.

2.6 Cluster correction using Gaussian Discriminant Analysis

The second stage of the algorithm is Gaussian Discriminant Analysis (GDA) that formalizes the genotype clusters obtained from the RF step and potentially improves the precision.

Let $D = \{x_j | j = 1 \dots N, x_j \in \mathbb{R} \times \mathbb{R} \times \mathbb{G} \times \hat{\mathbb{L}}\}$ denote a set of N validation SNPs that were classified by the trained Random Forest, where $\mathbb{G} = \{\mathrm{AA}, \mathrm{AB}, \mathrm{BB}\}$, $\hat{\mathbb{L}} = \{\mathrm{T}, \mathrm{F}\}$, Therefore, $x_j = (m_j, a_j, g_j, \hat{l}_j)$ is a quadruplet of the logarithmic difference, logarithmic average, genotype predicted by GenCall and class prediction by RF at the j-th SNP. We assume that both positive (T), and negative (F) class that are represented by pairs $d_i = (m_i, a_i)$ come from mixtures of multivariate normal distributions. Based on this, we define the following system of Gaussian discriminants:

$$\hat{L} \sim Bernoulli(\lambda) \quad (1)$$

$$p(\hat{l}) = \lambda^{\hat{l}}(1-\lambda)^{1-\hat{l}} \quad (2)$$

$$p(d_j|\hat{l} = T) = p(d_j|\Theta_{\mathrm{T}}) = \sum_{k=1}^{3} \alpha_{T,k}\phi(d_j|z_{T,k}, \theta_{T,k}) \quad (3)$$

$$p(d_j|\hat{l} = F) = p(d_j|\Theta_{\mathrm{F}}) = \sum_{k=1}^{3} \alpha_{F,k}\phi(d_j|z_{F,k}, \theta_{F,k}) \quad (4)$$

Where:

- λ denotes probability of $P(\hat{l} = T|d_j)$
- ϕ is multivariate normal density function with parameters θ_k (which is mean $\mu_k$and covariance matrix Σ_{k})
- z_k is indicator variable that denotes the genotype class, whereas $z_k \in \mathbb{G} \times \hat{\mathbb{L}}$
- α_k is the mixture component weight representing the probability that a random tuple (m_j, a_j) was generated by component k

The complete set of parameters for the presented Gaussian discriminants is given as:

$$\Theta_{\hat{\mathbb{L}} \in \{T,F\}} = \{\alpha_{\hat{\mathbb{L}},1}, \dots \alpha_{\hat{\mathbb{L}},3}, \theta_{\hat{\mathbb{L}},1} \dots \theta_{\hat{\mathbb{L}},3}\} \quad (5)$$

Decomposing the definition of z_k, we obtain following list of all components lying in two mixture models:

- a cluster of true heterozygous SNPs ($\mathrm{AB_{TRUE}}$)
- a cluster of false heterozygous SNPs ($\mathrm{AB_{FALSE}}$)
- a cluster of true homozygous SNPs ($\mathrm{AA_{TRUE}}$)
- a cluster of false homozygous SNPs ($\mathrm{AA_{FALSE}}$)
- a cluster of true homozygous SNPs ($\mathrm{BB_{TRUE}}$)
- a cluster of false homozygous SNPs ($\mathrm{BB_{FALSE}}$)

To estimate the parameters Θ_{L} we use maximum likelihood. The log-likelihood function $\mathcal{F}$ for classes from $\hat{\mathbb{L}}$ is defined as follows:

$$\ln \mathcal{F}(\Theta) = \sum_{j=1}^{N} \ln p(d_j|\Theta_{\hat{L}}) \quad (6)$$

In the prototype version of SureTypeSC we use Expectation Maximization algorithm (Dempster et al., 1977) to estimate the parameters $\Theta_{\hat{\mathrm{L}}}$ of positive and negative class that maximize their log-likelihood function (Eq. 6).

The EM algorithm is divided into Expectation-Step (E-Step) and Maximization-Step (M-Step). These are run in iterations until convergence is reached. N_l is total number of SNPs in the particular class.

Algorithm:

For every SNP i with label $\hat{l}$ in$\hat{\mathbb{L}}$:

1. E-step - calculate membership weights - probabilities of $(\mathrm{m_i,a_i})$ belonging to a cluster k using either initialization parameters $\Theta_{\hat{l}}^{init}$ if this is the first iteration, otherwise $\Theta_{\hat{l}}^{t+1}$

$$\omega_{i,k,\hat{l}} = p\left(z_{i,k,\hat{l}} = 1 \middle| d_i, \Theta_{\hat{l}}^{t}\right) = \frac{\Phi\left(d_i|z_{i,k,\hat{l}}\theta_{k,\hat{l}}\right) \cdot \alpha_{k,\hat{l}}}{\sum_{m=1}^{K} \Phi\left(d_i|z_{i,m,\hat{l}}, \theta_{m,\hat{l}}\right) \cdot \alpha_{m,\hat{l}}} \quad (7)$$

for $1 \le k \le 3, 1 \le i \le N$.

2. M-step
   - Calculate new component weights for the next iteration

$$\alpha_{k,\hat{l}}^{t+1} = \frac{\sum_{i=1}^{N_{\hat{l}}} \omega_{i,k,\hat{l}}}{N_{\hat{l}}} \quad (8)$$

- calculate new means for the next iteration

$$\mu_{k,\hat{l}}^{t+1} = \frac{\sum_{i=1}^{N_{\hat{l}}} \omega_{i,k,\hat{l}} d_i}{\sum_{i=1}^{N_{\hat{l}}} \omega_{i,k,\hat{l}}} \quad (9)$$

- calculate new covariances for the next iteration:

$$\Sigma_{k,\hat{l}}^{t+1} = \frac{\sum_{i=1}^{N_{\hat{l}}} w_{i,k,\hat{l}} \cdot (d_i - \mu_{k,\hat{l}}^{t+1})(d_i - \mu_{k,\hat{l}}^{t+1})^T}{N_{k,\hat{l}}} \quad (10)$$

at the end of the M-step we obtain new parameter estimates Θ^{t+1}

3. Calculate log likelihood using Equation 6 and if the relative change in the overall likelihood is smaller than a threshold, halt. Otherwise proceed with the E-step with parameters from Θ^{t+1}.

Note that we use two different categories of weights - membership weights (ω) and component weights (α). Membership weights are related to a particular SNP, whereas component weights represent the relative call rate of a particular genotype. After the parameters of both classes have been estimated by the EM algorithm, they are subjected to a second run. Here, the class membership $\hat{L}$ is hidden from the algorithm and every SNP is evaluated for both Gaussian discriminants using the following formula:

$$(\text{score}_T, score_F) = \ln p(d_j|\Theta_{\hat{L}}) \quad (11)$$

The final classification (membership to a positive or a negative class) is determined by higher value from the pair $(\text{score}_T, score_F)$. An example of a division of the feature space consisting of m and a by an EM algorithm is shown in Figure 1E. The SNPs were labeled according to the winning likelihood score and highest membership weight to a particular component of the winning class. In the implementation of SureTypeSC, we used variational Bayesian estimation algorithm with Dirichlet process (Blei and Jordan 2006) for the parameter inference implemented in scikit (Pedregoza et al., 2014). This allowed us to infer the number of actual components per class directly from the data and only give an input on maximum number of components per class ($k = 3$). In case of a haploid single cell sample this would lead to the weight of the heterozygous component being close to zero.

2.7 Scoring function

The key role of a genotyping algorithm is to report the likelihood of a certain genotype in form of a score or a posterior probability. Besides the GenCall having its own scoring scheme, we propose following equations to estimate the probability of a certain SNP being correctly genotyped:

1. Random Forest: the probability of a correct genotype of the *i*th SNP $P_{rf}^{i}(TRUE|d_i)$ is given as a proportion of the trees in the forest that voted for a particular genotype being correct
2. SureTypeSC: we apply Bayes rule to express the class-conditional posterior probability of a genotype falling into positive class T

$$score_{ST} = \frac{score_T + \ln p(\text{T})}{\sum_{z\in\{T,F\}} score_Z + \ln p(Z)} \quad (12)$$

3 Results

3.1 Generation of 28.7 million high confidence SNPs from single cells

We typed nearly 28.7 million SNPs from 104 cells from two individuals (GM12878 and GM07228) using the HumanKaryoMap-12 array (Illumina Inc., California, USA). To amplify the DNA from the single cells, we used multiple displacement amplification (MDA), a first-generation WGA method that is commonly used and relies on Phi (Φ) 29 polymerase. Its 3'→5' activity allows proofreading and therefore improves the fidelity of amplification. This allows high precision genotyping with a mutation rate of $10^{-7} - 10^{-9}$. Furthermore, the ability to displace secondary DNA structures, such as hairpin loops that would cause other polymerases to stall or dissociate from the template DNA, allows the amplification of long DNA fragments (2-10 kb).

3.2 Noise characterization of genotypes from single cells

To characterize the noise associated with genotyping from whole-genome amplified DNA from single cells, we compared the 28.7 million SNP genotypes from the two single cell datasets to their reference genotypes obtained from bulk, genomic DNA. To this end, we created high confidence reference genotypes from bulk DNA using nine independent gDNA samples hybridized against the HumanKaryomap-12 array and inferred genotypes using either the full parental information (GM07228, Supplemental Table 1) or multiple technical replicates of bulk DNA and sequence data (GM12878, Supplemental Methods and Eberle et al., 2017). This allowed us to identify 272,640 SNPs (98.9% autosomal SNPs) on the HumanKaryomap-12 array that called correctly in every replicate from bulk DNA. Using the standard QC cutoff from GenCall (0.15), 20.9 million SNPs from the two single-cell datasets called correctly, whereas 2.74 million generated false positives in GenCall, resulting in an incorrect genotype. 5.05 million SNPs gave 'no calls' (Table 1), having failed to fall within genotype clusters defined by bulk, DNA genotypes (a visualization of genotype clusters for one SNP is in Fig. 1A). The true positive rate was higher when we used a minimal QC (0.01) compared to the standard QC (44% and 40%, respectively, for cell line GM07228 and 36% to 33% for GM12878, Table 1, Supplemental Table 4). This suggests that the GenCall algorithm rejects about 7% correct genotypes from WGA DNA from single cells.

We displayed the pattern of the noise from the genotyping of SNPs from bulk and WGA DNA from single cells by first transforming the fluorescence intensities (X and Y) of each SNP into the logarithmic difference m and logarithmic average a (MA plot; Fig. 1B, C). The patterns of correctly called SNPs are similar between genotypes obtained from bulk or WGA DNA from single cells (blue contours, Fig. 1B, C). However, the noise distribution differed in several critical aspects. Three clusters of miscalls (false positives) became apparent in the single-cell data. Two clusters were from allele drop out (ADO), where AB genotypes were incorrectly genotyped as AA or BB. A smaller cluster of allele drop in (ADI) also appeared. The ADI cluster was clearly separated from the true AB genotypes. Most of the errors, however, occur in the transition area between AB to AA or AB to BB (ADO) but nevertheless suggest good separability of the correct calls from miscalls, since the centers of the clusters are non-overlapping (Fig. 1C).

Table 1. Summary of genotype calls from single cells

GM07228				
	Region	**Positive**[c]	**Negative**[d]	**No calls**[e]
Minimal QC[a]	AB	2,507,149	218,684	1,186,470
	AA,BB	10,139,650	1,939,749	
	all	12,646,799	2,158,433	
Standard Illumina QC[b]	AB	1,762,134	108,607	2,830,096
	AA,BB	9,794,074	1,496,791	
	all	11,556,208	1,605,398	

GM12878				
	Region	**Positive**[c]	**Negative**[d]	**No calls**[e]
Minimal QC[a]	AB	1,835,293	108,269	920,424
	AA,BB	8,443,004	1,413,298	
	all	10,278,297	1,521,567	
Standard Illumina QC[b]	AB	1,258,426	44,237	2,219,787
	AA,BB	8,106,969	1,090,869	
	all	9,365,395	1,135,106	

[a]GenCall cutoff 0.01
[b]GenCall cutoff 0.15
[c]concordant with the reference genotype
[d]not concordant with the reference genotype
[e]calls that did not meet the defined threshold

3.3 Design and implementation of the SureTypeSC algorithm

The characterization of the patterns of noise in a comprehensive dataset allowed us to employ a supervised machine learning method to classify and separate high quality genotypes from miscalls in the WGA DNA from single cells (Fig. S1). We combined a non-parametric (Random Forest) and parametric method (Gaussian mixture model) and developed a scoring strategy that assigns probabilities that a specific SNP from a single-cell dataset has been correctly genotyped (Methods, Eq. 5). The approach of using a Random Forest prevents over-fitting of the data and provides good estimates of the positive and negative class for the Gaussian discriminant analysis (Methods). We implemented the RF-GDA, termed SureTypeSC, and the testing procedures in Python using the scikit library (Pedregoza et al., 2014) and pandas (McKinney, 2010). SureTypeSC's input is compatible with GenomeStudio and allows the user to import the results of the analysis back to GenomeStudio for further investigation.

3.4 Validation of SureTypeSC on cross-validated data

To assess whether our algorithm captures noise from the single cell SNPs after undergoing WGA and to exclude the possibility of overtraining, where the model fits well to the training data but is not capable of generalizing to independent, unseen data, we first ran stratified 10-fold cross-validation on the single-cell dataset from cell line GM12878. The stratification ensures the same proportions of correctly genotyped SNPs and miscalls in all folds or samples. This is critical as the data are highly unbalanced, *i.e.* there are always more correctly genotyped SNPs than miscalls (Table 1). We trained the algorithm on 90% of the data and tested the performance of the remaining 10%. Due to stratification such that miscalls and correct calls were equally represented, in effect the training datasets contained 11,448,258-11,448,260 SNPs. As shown in Table 2, both the Random Forest on its own and SureTypeSC (the Random Forest in combination with the GDA) showed minimal deviations between the folds. This implies that the algorithms are invariant to SNP selection, since we sampled the SNPs randomly. The RF performs well across a range of measures, whereas SureTypeSC performs exceeding well for precision but at the expense of recall.

3.5 Validation of SureTypeSC on an independent dataset

We next addressed how well SureTypeSC or RF alone performed on an independent dataset of WGA data. To this end, we used the SNP genotypes obtained from 58 single cells after WGA from cell line GM07228 for training and the SNP genotypes obtained from WGA DNA from 46 single cells from a different cell line, GM12878 (Table 1), for testing ('tester set'). The genotyping data from the tester set were obtained at an independent time, with different batches of WGA reactions and genotyping arrays. This avoids systematic errors introduced by the chemistry used to obtain the genotypes. We evaluated the performance separately for heterozygous SNPs, homozygous SNPs and then for all SNP genotypes (Table 3). The results suggest that the RF alone can correctly genotype 91% of the heterozygous sites, which is about 20% greater than GenCall, whilst retaining a similar precision (96%). In contrast, the SureTypeSC algorithm consisting of both the RF and the GDA improved precision (99%) of the heterozygous sites, while still improving recall by 4 % compared to GenCall (Table 3). The ROC curves were shifted to the left for both the RF and SureTypeSC (Figure 2A) and the ROC-AUC (area-under-the-curve) was also increased after implementation of the two different algorithms compared to GenCall (Table 3).

Table 2. Results of the cross-fold validation on dataset GM12878[a]

Algorithm \ Metrics	precision	recall	accuracy	f1 score
RF	0.92 ± 0.001	0.95 ± 0.003	0.88 ± 0.002	0.93 ± 0.001
SureTypeSC[b]	0.95 ± 0.001	0.87 ± 0.01	0.85 ± 0.007	0.90 ± 0.005

[a]the dataset was subjected to minimal QC (GenCall score cutoff 0.01) and then randomly split into 10 subsets, 9 subsets were used for training and the rest was used for testing, this procedure was repeated 10 times, then the average and standard deviation was calculated; results are displayed as proportions
[b]SureTypeSC combines RF and GDA.

Table 3. Performance of tested classifiers on GM12878[a]

Metrics \ Algorithm	GenCall[b]			RF[c]			SureTypeSC[d]		
	het	homo	all	het	homo	all	het	homo	all
precision	0.97	0.88	0.89	0.96	0.89	0.9	0.99	0.93	0.94
recall	0.69	0.96	0.91	0.91	0.97	0.96	0.73	0.9	0.87
f1-score	0.8	0.92	0.9	0.94	0.93	0.93	0.84	0.92	0.9
roc-auc score	0.71	0.66	0.64	0.8	0.77	0.77	0.87	0.83	0.81

[a]data subjected to minimal QC with GenCall cutoff 0.01; values are proportions
[b]GenCall score cutoff 0.15
[c]RandomForest score cutoff 0.5
[d]SureTypeSC is combination of RF and GDA with score cutoff 0.9

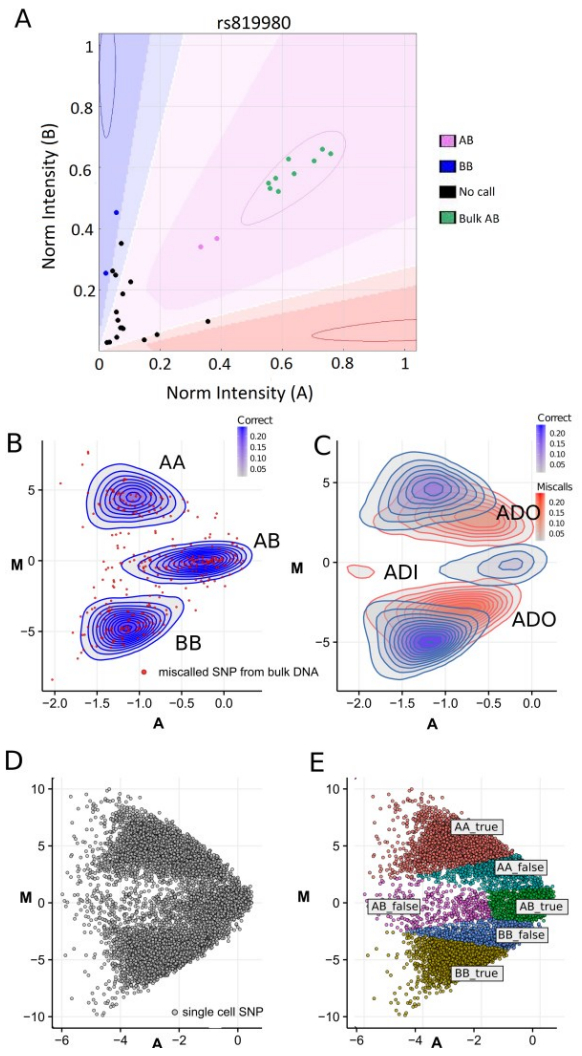

Fig. 1. Signal-noise detection in whole-genome amplified DNA from single cells. (A) The GenCall algorithm in GenomeStudio classifies genotyping calls based on the normalized intensities of the X and Y channels (A and B allele, respectively). The genotyping space for homozygous AA calls is shown in blue, heterozygous genotypes fall within the purple area and homozygous BB genotypes are in red. The white lines represent the absolute cut-offs for distinct genotyping areas, whereas the lighter shades around the main genotyping areas represent lower confidence areas, where the specific SNP was correctly typed, but fell below the QC threshold of 0.15. The centroid of each genotyping space is shown as a circle. The genotyping space is specific to each SNP and based on bulk DNA. Green points: genotypes from bulk DNA, purple points: correct genotypes from single cell, black points: genotyping calls from single cells below the QC threshold of GenCall; blue points: misclassified genotype from single cells. (B) Contour MA plot of all SNPs from one bulk DNA sample (gDNA-01) from GM07228; AA, BB and AB clusters are labeled accordingly. Correctly typed SNPs are rendered in blue, whereas incorrectly typed SNPs are shown in red (C) Contour MA plot of all SNPs from one single-cell sample (sc-21) from GM07228; AA, BB and AB clusters are labeled accordingly. (D) MA plot of 10,000 randomly selected SNPs from 10 single cell samples from GM07228. (E) Cluster labeling of 10,000 randomly selected SNPs from (D)

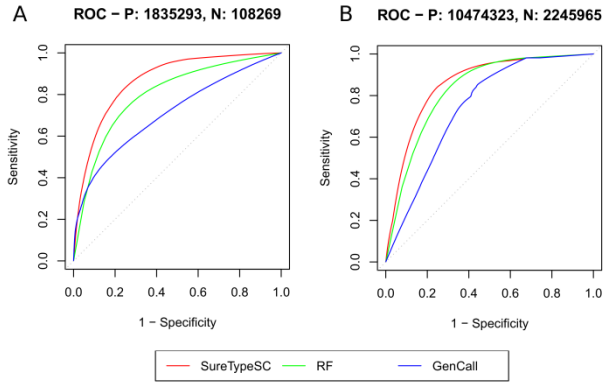

Fig. 2 SureTypeSC improves the performance for single cell genotyping. ROC curve for heterozygous (A) and homozygous SNPs (B). P and N are the numbers of correctly (P, positive) and incorrectly (N, negative) typed SNPs. The dotted line in the ROC curves is the diagonal (random classification).

For homozygous genotypes, the RF and SureTypeSC also shifted the ROC curves to the left (Figure 2B) and increased the ROC-AUC score (Table 3). The SureTypeSC improved precision compared to both GenCall and the RF alone (93%), but at the expense of nearly 7% fewer typed SNPs (recall, 88%). Collectively our results show that improved precision can be obtained when genotyping both heterozygous and homozygous SNPs from WGA DNA by using our algorithms.

3.6 Call confidence in the single-cell environment.

Our observations that recall of heterozygous SNPs increases from 69% to 91% suggest that the RF retains a much larger proportion of heterozygous SNPs that are rejected in GenCall. Furthermore, precision is improved to 99% in SureTypeSC, at a similar recall rate as GenCall (Table 3). The precision can be viewed as a confidence metrics that a particular genotype is called correctly. We extended this approach and developed a statistical toolkit that shows a detailed view of confidence in AA, BB or AB calls using a transition matrix consisting of posterior probabilities. The posterior probabilities (elements of Table 4) show the probability that a certain genotype from the single cell application (SC in column) is being called correctly as reference genotype (Ref, row). To demonstrate this, we generated genotypes from our single cell datasets (GM12878) using GenCall (minimal QC 0.01) and then applied GenCall with a QC 0.15 or SureTypeSC (with a SureTypeSC threshold of 0.9). Table 4 shows that we improved the confidence in single-cell AA and BB calls by 7% (or by 5% compared with standard GenCall genotyping), respectively. The confidence in single-cell AB calls is concordant with the precision obtained in the validation analysis in Table 3 (99%). We were furthermore interested whether we could improve precision with our algorithm after genotyping with standard GenCall parameters (QC 0.15). Supplemental Table 2 shows that this configuration still outperforms GenCall QC 0.15 but gives lower precisions for homozygous regions than SureTypeSC with GenCall QC 0.01 (Table 4). This suggests that the specificity of GenCall is suboptimal when applying the standard thresholding framework in the single cell environment. This is further supported by the ROC curve (Figure 2).

Table 4. Precision rates with GenCall and SureTypeSC on single cell line from GM12878[a]

Minimal QC[b]					
Ref \ SC	AA	AB	BB	NC	Call rate
AA	**0.85**	0.008	4.2×10^{-5}	0.09	0.37
AB	0.15	**0.94**	0.13	0.61	0.15
BB	0.0001	0.05	**0.87**	0.08	0.41
NC	0.004	0.009	0.005	**0.22**	0.07
GenCall[c]					
Ref \ SC	AA	AB	BB	NC	Call rate
AA	**0.87**	0.003	1.7×10^{-5}	0.12	0.34
AB	0.12	**0.96**	0.11	0.66	0.1
BB	4.4×10^{-5}	0.03	**0.89**	0.12	0.38
NC	0.004	0.009	0.005	**0.1**	0.17
SureTypeSC[d]					
Ref \ SC	AA	AB	BB	NC	Call rate
AA	**0.92**	0.002	1.8×10^{-5}	0.13	0.31
AB	0.08	**0.99**	0.06	0.59	0.11
BB	1.85×10^{-5}	0.005	**0.94**	0.2	0.34
NC	0.003	0.008	0.003	**0.07**	0.25

[a]elements of the table show confidence (precision) rates of a particular SC genotype (column) being genotyped as in reference (row), the last column shows the call rates in the single cell data
[b]GenCall score cutoff 0.01
[c]GenCall score cutoff 0.15
[d]GenCall score cutoff 0.01 and SureTypeSC score cutoff 0.9

3.7 Allele-drop out and allele-drop in rates are reduced using SureType SC

Incorrect genotype calls arise predominantly from imbalances in the allele frequencies during the chemical reaction when the whole genome is amplified. The deviation from a 1:1 allele ratio of heterozygous SNPs can lead to allele drop out (ADO). Analogously, mistyping of a homozygous SNP results in allele drop in (ADI). We estimated the ADO and ADI rates before and after implementation of the Random Forest and SureTypeSC algorithm using the transition matrices of the posterior probabilities (Table 4; Methods).

Table 5 shows that the RF and GenCall had similar ADI and ADO rates (3% and 10-12%, respectively), with similar call rates. In contrast, ADI rate was reduced from 5% to 0.7% and the ADO rate decreased from 14% to 7% after SureTypeSC. Thus, SureTypeSC significantly reduces ADO and ADI rates, but at a cost of the call rate, which was decreased from 83% (QC 0.15) to 75%. Thus, the improvement in correct genotype detection in the single-cell environment comes with a certain data loss.

Table 5. Allele drop-in, allele drop-out and call rate with GenCall and SureTypeSC

	Min. QC[a]	GenCall[b]	RF[a,c]	SureTypeSC[a,d]
ADI	0.05	0.03	0.03	0.007
ADO	0.14	0.12	0.1	0.07
Call rate	0.93	0.83	0.85	0.75

[a]GenCall score cutoff 0.01
[b]GenCall score cutoff 0.15
[c]RandomForest score cutoff 0.5
[d]SuretypeSC score cutoff 0.9
[e]SuretypeSC score cutoff 0.9

Discussion

In this study, we have typed nearly 30 million SNPs from 104 single cells from two independent cell lines and developed an algorithm to distinguish signal from noise in whole-genome amplified DNA. The algorithm has two layers that are organized in a cascade. The first layer of SureTypeSC accepts nearly all SNPs (QC of 0.01) in the dataset. This helps the Random Forest in the first layer to learn the full error pattern from the single-cell data. As Table 3 suggests this improves the rate of correctly identified SNPs, since recall increased from 69% (GenCall) to 91% for heterozygous SNPs. Resolving most of the heterozygous SNPs makes the first layer highly relevant and applicable when heterozygosities are needed, such as tag SNPs during linkage analysis of transmission of monogenic diseases and aneuploidy detection. The second layer of the SureTypeSC algorithm is a system of two GMMs, which aims to maximize the precision at cost of potential data loss. Having high precision makes it feasible to explore rare events across populations of cells. This includes assessing clonal expansion in tumour evolution, linage tracing, or detecting rare *de novo* mutations in single cells that are averaged out and lost in bulk analyses (Cooper et al., 2015; Lu et al., 2012; Chen et al., 2017).

Analysing a large number of single cells allows the decomposition of heterogeneous populations. Furthermore, having a robust algorithm of genotyping from WGA DNA from single cells improves the certainty of genotype calling when only few cells are available. This is important in both basic biomedical research as well as clinical settings such as in preimplantation genetic testing. Whereas there are specialised tools for single cell genotyping from next-generation sequencing (Zafar et al., 2016) that improve precision, in the field of single-cell SNP array analysis this has been only achieved by systematic increase of the genotyping algorithms's thresholds and causes a substantial data loss (Zamani et al., 2015). Although genotyping from SNP arrays cover only a fraction of the genome compared to next-generation sequencing, the cost of *de novo* genome assembly is prohibitive even for bulk, genomic DNA when assessing a large number of cells. The sequencing depth, or coverage, needed in one recent reference genome assembly for the detection of *de novo* mutations was nearly 50× (Besenbacher, 2015). For single-cell applications, the coverage to accurately identify new mutations from the noise and bias introduced by the whole-genome amplification step is in excess of this (Behjati et al., 2014). Thus, SureTypeSC allows a cost-effective approach to improving genotype precision using SNP arrays.

Acknowledgements

The following cell lines/DNA samples were obtained from the NIGMS Human Genetic Cell Repository at the Coriell Institute for Medical Research: GM07228, GM07224, GM07225 and GM12878. We thank Alex Bladon from Illumina for critical discussion and members of Hoffmann lab for proofreading of the manuscript.

Funding

I.V. was supported by the Danish National Research Foundation Center grant (DNRF115). Work in the ERH lab was supported by a NNF Young Investigator Award (16662).

Conflict of Interest: The authors declare no conflict of interest for developing a single-cell genotyping algorithm. For transparency, this study uses Illumina products, because R. Blanshard is employed by Illumina UK Ltd. E. R. Hoffmann receives funding in-kind from Illumina UK Ltd. R. Blanshard is a registered Ph.D. student at the University of Sussex, UK, under the supervision of E. R. Hoffmann.

References

Behjati, S., Huch, M., van Boxtel, R., Karthaus, W., Wedge, D.C., Tamuri, A.U., Martincorena, I., Petljak, M., Alexandrov, L.B., Gundem, G., et al. (2014). Genome sequencing of normal cells reveals developmental lineages and mutational processes. Nature 513, 422–425.

Bland, J. M. and Altman, D. G. (1999). Measuring agreement in method comparison studies. Statistical Methods in Medical Research, 8(2), 135–160.

Blanshard, R.C., Chen, C., Xie, X.S., and Hoffmann, E.R. (2018). Chapter 20 - Single cell genomics to study DNA and chromosome changes in human gametes and embryos. In Methods in Cell Biology, H. Maiato, and M. Schuh, eds. (Academic Press), pp. 441–457.

Blei, D.M., and Jordan, M.I. (2006). Variational Inference for Dirichlet Process Mixtures.

Breiman, L. (2001). Random forests. Mach. Learn., 45(1), 5–32.

Capalbo, A., Ubaldi, F. M., Rienzi, L., Scott, R., and Treff, N. (2017). Detecting mosaicism in trophectoderm biopsies: current challenges and future possibilities. Human Reproduction (Oxford, England), 32(3), 492–498.

Carvalho, B., Bengtsson, H., Speed, T. P., and Irizarry, R. A. (2007). Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. Biostatistics (Oxford, England), 8(2), 485–499.

Chen, C., Xing, D., Tan, L., Li, H., Zhou, G., Huang, L., and Xie, X. S. (2017). Single- cell whole-genome analyses by Linear Amplification via Transposon Insertion (LIANTI). Science (New York, N.Y.), 356(6334), 189–194.

Cooper, C.S., Eeles, R., Wedge, D.C., Van Loo, P., Gundem, G., Alexandrov, L.B., Kremeyer, B., Butler, A., Lynch, A.G., Camacho, N., et al. (2015). Analysis of the genetic phylogeny of multifocal prostate cancer identifies multiple independent clonal expansions in neoplastic and morphologically normal prostate tissue. Nat. Genet. 47, 367–372.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B. Methodological, 39(1), 1–38.

Eberle, M.A., Fritzilas, E., Krusche, P., Källberg, M., Moore, B.L., Bekritsky, M.A., Iqbal, Z., Chuang, H.-Y., Humphray, S.J., Halpern, A.L., et al. (2017). A reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. Genome Res *27*, 157–164.

Gawad, C., Koh, W., and Quake, S.R. (2016). Single-cell genome sequencing: current state of the science. Nat Rev Genet 17, 175–188.

Giannoulatou, E., Yau, C., Colella, S., Ragoussis, J., and Holmes, C.C. (2008). GenoSNP: a variational Bayes within-sample SNP genotyping algorithm that does not require a reference population. Bioinformatics 24, 2209–2214.

Kermani BG (2008). Artificial intelligence and global normalization methods for genotyping.

Illumina, Inc. (2014). Infinium Genotyping Data Analysis. page 10.

International HapMap Consortium (2003). The International HapMap Project. Nature, 426(6968), 789–796.

International HapMap Consortium. A haplotype map of the human genome. Nature. 2005 Oct 27;437(7063):1299-320.

Li, G., Gelernter, J., Kranzler, H.R., and Zhao, H. (2012). M(3): an improved SNP calling algorithm for Illumina BeadArray data. Bioinformatics 28, 358–365.

Lu, S., Zong, C., Fan, W., Yang, M., Li, J., Chapman, A. R., Zhu, P., Hu, X., Xu, L.,Yan, L., Bai, F., Qiao, J., Tang, F., Li, R., and Xie, X. S. (2012). Probing meiotic recombination and aneuploidy of single sperm cells by whole-genome sequencing. Science (New York, N.Y.), 338(6114), 1627–1630.

McKinney, W. (2010). Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference 51–56.

Oshiro, T. M., Perez, P. S., and Baranauskas, J. A. (2012). How Many Trees in a Random Forest? In Machine Learning and Data Mining in Pattern Recognition, Lecture Notes in Computer Science, pages 154–168. Springer, Berlin, Heidelberg.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Ritchie, M. E., Liu, R., Carvalho, B. S., and Irizarry, R. A. (2011). Comparing genotyping algorithms for Illumina's Infinium whole-genome SNP BeadChips. BMC Bioinformatics, 12, 68.

Ritchie, M.E., Carvalho, B.S., Hetrick, K.N., Tavaré, S., and Irizarry, R.A. (2009). R/Bioconductor software for Illumina's Infinium whole-genome genotyping BeadChips. Bioinformatics 25, 2621–2623.

Saito, T., Rehmsmeier, M., and Wren, J. (2017). Precrec: fast and accurate precision–recall and ROC curve calculations in R. Bioinformatics 33, 145–147.

Teo, Y.Y., Inouye, M., Small, K.S., Gwilliam, R., Deloukas, P., Kwiatkowski, D.P., and Clark, T.G. (2007). A genotype calling algorithm for the Illumina BeadArray platform. Bioinformatics 23, 2741–2746.

Zafar, H., Wang, Y., Nakhleh, L., Navin, N., and Chen, K. (2016). Monovar: single-nucleotide variant detection in single cells. Nature Methods 13, 505–507.

Zamani Esteki, M., Dimitriadou, E., Mateiu, L., Melotte, C., Van der Aa, N., Kumar, P., Das, R., Theunis, K., Cheng, J., Legius, E., et al. (2015). Concurrent whole-genome haplotyping and copy-number profiling of single cells. Am. J. Hum. Genet. 96, 894–912.