BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Faculty of Information Technology
Fakulta informačních technologií

PHD THESIS SUMMARY
TEZE DISERTAČNÍ PRÁCE

Brno, 2019 Ivan Vogel

Keywords

genotyping, bulk DNA, single cell, machine learning, discriminant analysis, recombination, SNP array

Klíčová slova

genotypování, hromadné DNA, jediná bunka, strojové učení, diskriminační analýza, rekombinace, SNP mikročip

Table of contents

Table of contents.....1

1 Introduction.....3

  1.1 Motivation.....3

  1.2 Goals of the thesis.....4

    1.2.1 Research questions.....4

  1.3 Structure of this summary.....5

2 Genomics.....6

  2.1 Workflow for processing genomic data.....6

    2.1.1 WGA.....6

    2.1.2 Microarrays.....7

    2.1.3 Problem definition.....8

    2.1.4 Single-cell genotyping.....8

3 Novel algorithm for noise filtration.....10

  3.1 Introduction.....10

  3.2 Datasets of biological data.....10

    3.2.1 Groundtruth genotype from gDNA.....10

  3.3 Comparison of scDNA to gDNA.....10

  3.4 Structure of noise in single cell.....12

  3.5 Training dataset.....13

    3.5.1 Feature selection.....13

    3.5.2 Problem of imbalanced dataset.....13

  3.6 SureTypeSC.....14

    3.6.1 Basic Random Forest layer.....14

    3.6.2 Refinement using Gaussian Discriminant Analysis.....15

    3.6.3 Scoring function.....16

4 Validation.....17

  4.1 Validation curves and ROC-AUC score.....18

  4.2 Evaluation with fixed thresholds.....19

5 Knowledge extraction from single cell data.....21

  5.1 Data model of recombination in meiosis.....21

    5.1.1 Introduction.....21

    5.1.2 Meiotic pathways and data models.....21

    5.1.3 Model I – reconstructing crossovers from Duos.....22

    5.1.4 Model II – reconstructing crossovers from Trios.....23

5.2 Gene conversions 25
    5.2.1 Introduction 25
    5.2.2 Detection of GCs in Trios 26
5.3 Conclusion 26
6 Applications of SureTypeSC 27
6.1 Introduction 27
6.2 Improved crossover detection 27
    6.2.1 Duos 27
    6.2.2 Trios 27
6.3 Direct detection of gene conversions 29
    6.3.1 Genotyping with GenCall 29
    6.3.2 Filtering with SureTypeSC 30
6.4 Detection of copy number variants 31
6.5 Detecting subpopulations in the single cells 32
    6.5.1 Introduction 32
    6.5.2 Results 32
7 Conclusion 34
7.1 Summary of contribution 35
Bibliography 36
Curriculum Vitae 38
Abstract 41

1 Introduction

1.1 Motivation

It has been over half of the century since Watson and Crick discovered the molecular structure of deoxyribonucleic acid (DNA). DNA, the code of life, codes the genetic information that we inherited from our ancestors and will partly pass on to our offsprings. Every single cell within our body contains nearly the same genetic information. However, every single individual (as a system of multiple cells communicating with each other) contains a collection of single nucleotide polymorphisms (SNPs) that distinguishes him from the rest of the population. One of the crucial tasks of nowadays computational biology is to shed light on this genetic diversity. A process of revealing a particular SNP is called SNP-typing or genotyping. Genotyping is feasible thanks to powerful algorithms relying on robust statistical methods assuming sufficiently strong signal that supports a genetic variant in a population. Invariant to the screening technology used, we need to operate with sufficient amount of biological material to obtain strong signal supporting our observation. This is usually not a problem, as, i. e., blood sample from an individual would likely contain sufficient amount of genetic material from multiple cells. Although all the cells within one individual should in theory contain the same genetic information, there is a plethora of factors related to their function that can alter the DNA dynamically during their lifetime – i.e. epigenetics or *de-novo* mutations (Junker and van Oudenaarden, 2014; Shapiro et al., 2013). Therefore, there is an increasing motivation constantly driven by new scientific discoveries to look at the DNA on a single cell level. One big driver are cancer studies, where we know that cancer cells are highly heterogeneous due to increased mutagenesis. Knowing the precise genetic structure of a cancer cell can be crucial when assessing the therapy. Another example, where analysis on single cell level is of a great benefit, is examining structure of DNA and biological processes in highly complex tissues i.e. brain. At last, but not least, reproductive biology is a field where knowing the precise genetic information of a single cell is crucial. During creation of reproductive cells in human, the process called recombination shuffles the genetic information. This shuffling is unique per cell and helps us understand potential genetic disorders transmitted to offspring, causes of miscarriages etc. A very practical example, that is possible thanks to advanced medical technology, is in-vitro fertilization. During this procedure, a set of female eggs is monitored and genetically screened and only the eggs with the best fitness (in terms of likelihood of genetic diseases) will be used for in-vitro fertilization.

When analyzing data from multiple cells in a pool (termed bulk DNA), we accept that the obtained signal is an average of all cells in the sample. We illustrated many examples, where this is not good enough. Thanks to sophisticated methods of molecular biology, we are now able to sort cells from a tissue and pick an individual cell and extract DNA. The amount of single cell DNA, unlike in bulk DNA is however very little for successful screening and therefore, the DNA has to be „copy-pasted". This process is possible, it is called whole-genome amplification, however, it deteriorates the signal and can compromise the whole analysis by introducing bias to the data (Spits et al., 2006; Vanneste et al., 2012).

Machine learning has been successfully applied to many interdisciplinary fields including bioinformatics, biomedical research and medicine. Algorithms and statistical methods from this field many times improved the state of the art biological model by providing valuable and accurate predictions in situations, where searching through the whole space of possible values would be simply too costful or fatal.

SNP array has been an affordable technology for screening variants in DNA of an individual for over a decade and a package of hardware-software solution for revealing the genotype has shown high accuracy when bulk DNA is analysed, but not the single cell (Vanneste et al., 2012).

This works gives on overview of strategies for bioinformatics analysis of genomic data and then particularly focuses on algorithms for bulk DNA and single-cell genotyping. It pinpoints the specificities of the single-cell environment compared to standard bulk DNA processing and it particularly researches the anatomy of noise and possibilities of improvement of genotyping. An original machine learning algorithm is proposed that tackles the noise problem. Additionally, few algorithms have been designed for single cell genomics, particularly area of reproductive biology, that, together with the novel machine learning solution, improve the current state of the art in bioinformatics of single cell and allow to look at fine genomic events at the single cell level.

1.2 Goals of the thesis

Given the motivation in the introduction, this work primarily focuses on investigating solutions for single-cell genotyping using data from SNP arrays. Based on the literature survey, there are satisfying solutions for genotyping the bulk DNA with SNP array, but none of them was specifically designed to genotype DNA from a single cell. Documented by multiple studies, the single cell data contains noise caused by amplification of the genetic material. This is a problem, because errors in the data can lead to false biological conclusions and therefore compromise the whole workflow. Standard genotyping algorithms fail, because they do not assume deteroriated signal caused by erroneous amplification. As a direct consequence, they both, allow high Type I and Type II errors.

1.2.1 Research questions

To solve the question of reliable genotyping of single cell data, we can formulate following research questions:

1. Is it possible to describe pattern of noise in the SNP array single-cell data?

2. Is is possible to design a machine learning method that would distinguish the good quality data from the noise and improve precision of a single-cell genotype?

1.3 Structure of this summary

During his doctoral studies, the author of this thesis was member of two research groups[1] and the results presented here are author’s contribution to their scientific outcome. The research groups actively supported the informatics research of the author with the exclusive data and this is accordingly reflected in the structure and the content of the work.

The next chapter will introduce the most important theoretical background of this work. As the single cell analysis shares many steps with standard analysis of bulk genomic data, we will also present generic workflow and pinpoint the specificities of the single cell environment. Chapter 3 is dedicated to the novel algorithm for single cell data filtration using machine learning called SureTypeSC. Chapter 4 then validates this approach and Chapter 5 demonstrates few newly designed knowledge extraction algorithms over single-cell data. Chapter 6 shows applications of SureTypeSC in various practical scnearios of biological inference. Chapter 7 summarizes the contribution and proposes ideas for future research.

---

[1] Institute of Biophysics of the Czech Academy of Sciences, Department of Plant Developmental Genetics in Brno, Czech Republic and Center for Chromosome Stability, Department of Cellular and Molecular Medicine, University of Copenhagen, Denmark

2 Genomics

„*Genomics is the study of the full genetic complement of an organism (the genome). It employs recombinant DNA, DNA sequencing methods, and bioinformatics to sequence, assemble, and analyse the structure and function of genomes.*“[2]

2.1 Workflow for processing genomic data

A generic workflow for processing genomic data is shown in Figure 2.1. The whole methodology can be dividied into two consecutive parts: `laboratory part` (Figure 2.1A) and `software part` (Figure 2.1B). The main interest of this work is the software part and will be discussed in more detail. From the laboratory part, (a) whole genome amplification (WGA), and (b) microarray technology are crucial for this work. WGA is a bottleneck of the state of the art single cell analysis as it has the greatest impact on the accuracy of the software part. Microarray an affordable technology for testing thousands of genomic position in parallel manner. The reader is refered to Vogel (2019). for details and references on the other steps of the laboratory part.

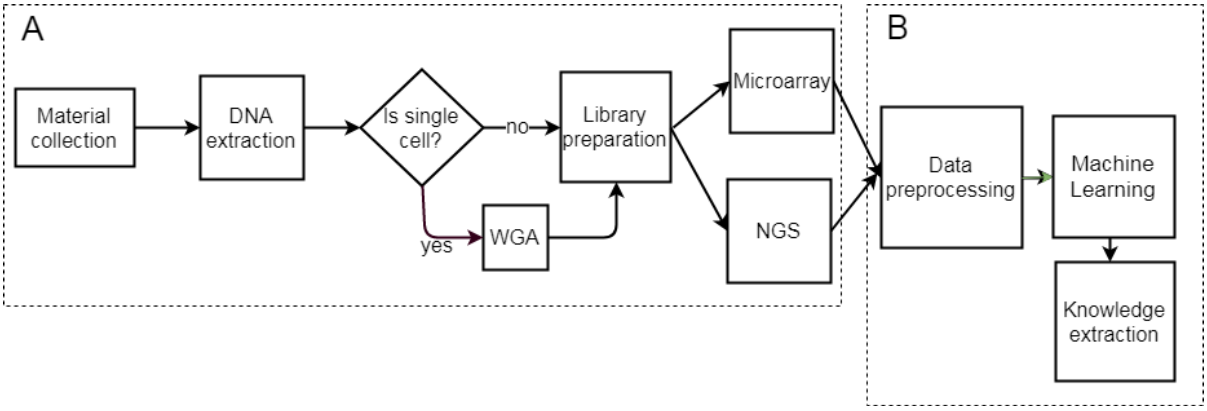


Figure 2.1. Workflow for processing of the genomics data. (A) Laboratory part starts with material collection and DNA extraction. Should the material be from single cell, whole genome amplification is required. Subsequently, library of genomic data is created and screened by two commonly used technologies – microarrays or next-generation sequencing. (B) The signal from both microarray or NGS is preprocessed and then, a machine learning model is created from the data to predict primary information (i.e. genotyping). Knowledge extraction then takes into account the genomic context and draws biological conclusions about the data (i.e. presence of recombination event)

2.1.1 WGA

WGA is a crucial step in the the single cell analysis (Figure 2.1) due to the critically low amount of DNA. The criterion for a good performing whole genome amplification method is amplification evenness and low error rate. Evenness refers to the ability to evenly cover the whole genome (Blanshard et al., 2018). While there is couple of amplification kits available on the market at the

---

[2] https://www.nature.com/subjects/genomics

moment (Blanshard et al., 2018), their common drawback is that they introduce bias to the data (Figure 2.2)

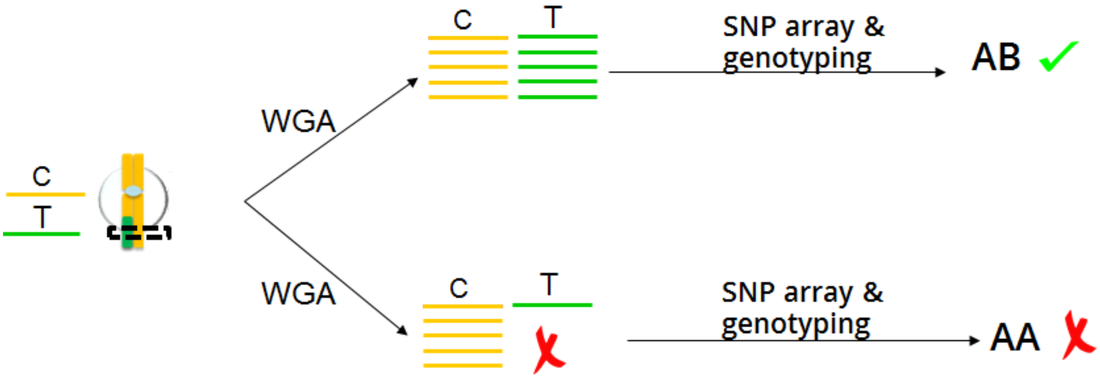


Figure 2.2. WGA – cause of errors in the single cell genotyping. The heterozygous locus in the dashed rectangle is correctly amplified with nearly equal amounts of both of the alleles (top) and then correctly genotyped as AB. Other situation is that, due to erroneous whole genome amplification, one of the alleles has suboptimal signal (bottom branch). This causes wrong detection of homozygous genotype.

2.1.2 Microarrays

DNA microarray, also known as biochips or DNA chips, is a technology that allows parallel measurement of genetic information using hybridization principle. Hybridization is a biochemical reaction, where two complementary DNA strands attach to each other by hydrogen bond to create duplex. The generic technology consists a set of `probes` and `targets`. Probes are distributed on a solid surface and contain short fragments of DNA of known sequence. Target is the DNA of interest. The target DNA is usually fragmented into smaller pieces and fluorescently labeled to enable the detection. Once in contact with the probes on microarray, the labeled fragments will specifically hybridize with the arranged probes and the signal is then measured and quantified (Figure 2.3).

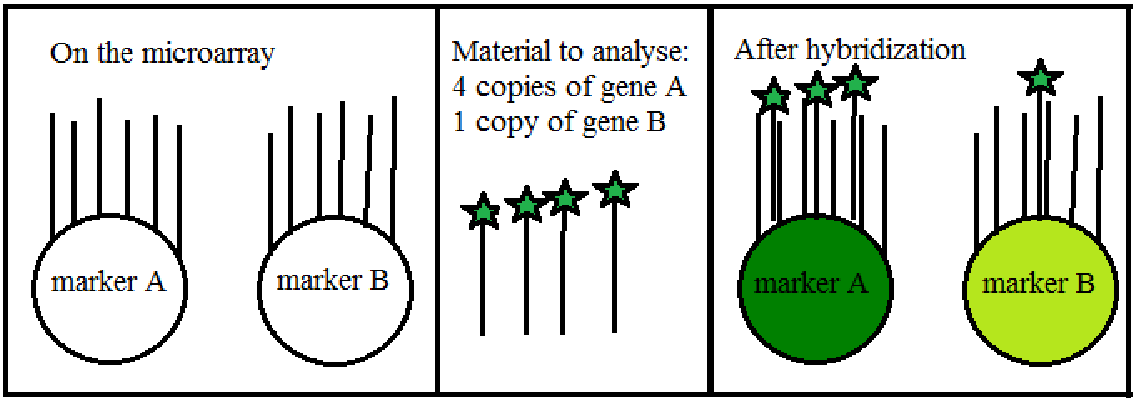


Figure 2.3. Principle of microarray. The microarray contains spots with short fragments attached to a bead (lines attached to marker A and marker B). Once hybridized with the fluorescently labeled material to analyse (target), the signal can be quantitatively assesed from the intensity of the fluorescence.

Illumina SNP array

Illumina platform called Illumina BeadChip (Illumina, Inc.), that will be exclusively presented in this work, is using universal probes of 50 nucleotides. The actual genotyping takes place one nucleotide after the probe using single nucleotide extension assay (Illumina Infinium Assay protocol, Illumina Inc.). Depending on the target SNP, the complementary nucleotide carries either red or green fluorescent signal (A, T or G,C, respectively). The output of the Illumina technology is therefore one raw measurement for the A allele and one for the B allele at each SNP. The number of SNPs and samples that can be analysed in parallel on one chip varies from 4 to 12 samples and from ~300k SNPs to 1 million SNPs depending on the specific product[3].

2.1.3 Problem definition

Every SNP i on an array is assigned a tuple of raw intensities d_i, defined as follows:

$$d_i = (x_i, y_i), \quad (3.1)$$

$$d_i \in A\mathbb{R}^2 \quad (3.2)$$

Where $A\mathbb{R}^2$ is affine space. Genotyping is a mapping function $\mathcal{F}$ in a multiclass classification problem $C = 3$

$$\mathcal{F}: \{d_i\}_{i=1}^{N} \rightarrow \{AA, BB, AB\} \quad (4.1)$$

To simplify the problem, we can put following fuzzy definitions over d_i:

$$x_i = high \ \wedge y_i = low \ \rightarrow AA \quad (3.1)$$

$$x_i = low \ \wedge y_i = low \ \rightarrow BB \quad (3.2)$$

$$x_i = y_i \rightarrow AB \quad (3.3)$$

2.1.4 Single-cell genotyping

Currently, there is couple of tools available for genotyping of SNP array data (Vogel, 2019). To the best of our knowledge, there is not a specialized algorithm available for the single cell genotyping from SNP arrays. Zamani Esteki et al. (2015) performed evaluation of two genotyping algorithms trained on bulk DNA in the single cell environment and attempted to adjust them to single cell data. An important summary of this is shown in Figure 2.4, where the authors analysed the homozygous and heterozygous genotypes separately. As both methods give a score or measure of confidence of a genotype, they were systematically increasing the thresholds to acchieve better accuracy. The results suggest that while GenoSNP (Giannoulatou et al., 2008) has generally higher call rate, it also suffers from lower accuracy compared to GenCall, particularly for the heterozygous calls. Zamani Esteki et al. decided to proceed with GenCall due to better overall properties. Adjusting the threshold of the GenCall algorithm to the single cell envorinment came at cost of significant data loss (call rate ~60 % at accuracy below 90 % , Figure 2.4). To give a frame of reference, validation studies on bulk DNA indicate both accuracy and call rate above 99% on average for both algorithms (Ritchie et al., 2011).

---

[3] Information and details about a particular SNP array technology available at www.illumina.com

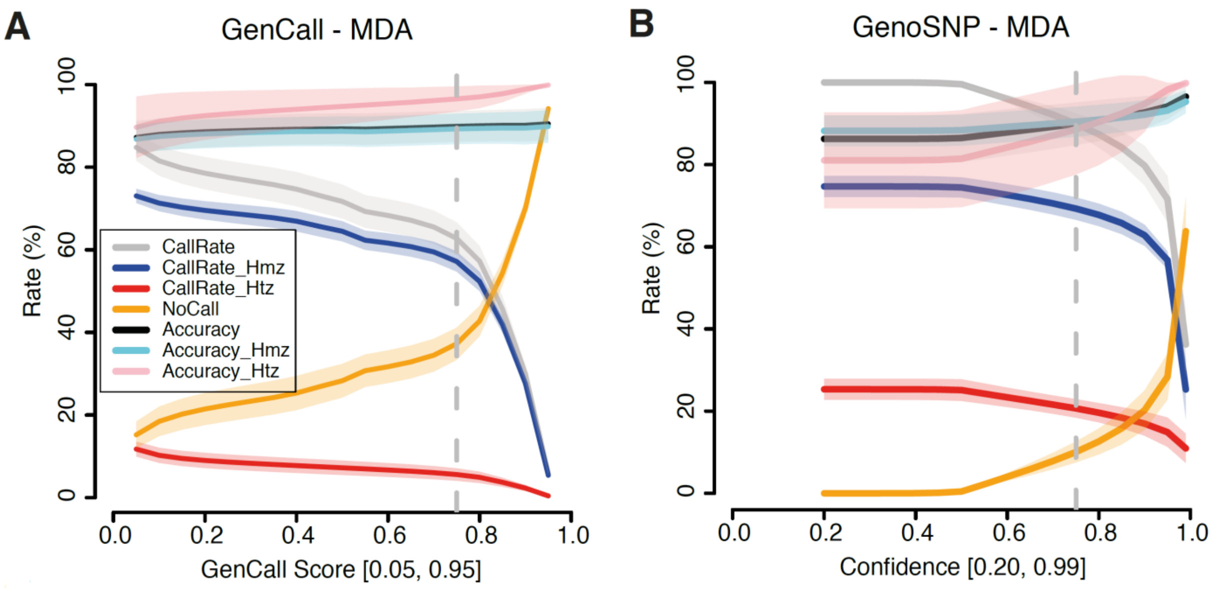


Figure 2.4. Comparison of performance of GenCall (A) and GenoSNP (B) on the single cell data amplified by multiple displacement amplification (MDA, technology of WGA). The dashed vertical line shows the actual threshold that was selected by Zamani Esteki et al.(2015) for the genotyping as tradeoff of accuraccy and call rate. Hmz means homozygous, Htz is heterozygous.

3 Novel algorithm for noise filtration

3.1 Introduction

This chapter presents original algorithm that removes noisy genotypes from single-cell data caused by whole genome amplification (WGA). It presents results from study published in Vogel et al. (2019). To understand the noise, it is crucial to define a reference population of single cells where we know the ground truth.

We operate with two categories of biological data. **Bulk genomic DNA (gDNA)** is characterized by sufficient amount of genetic material (pool), has great support in genotyping tools and gives very precise genotype estimation. On contrary, **single cell DNA (scDNA)** is characterized by very small amounts of DNA (from one cell) and needs to undergo a single cell path of the workflow for processing genomic data (Figure 2.1). This protocol causes deterioration of the signal and random erroneous genotypes at the output (Figure 2.2).

3.2 Datasets of biological data

We operated with two human cell lines, GM12878 and GM7228. For these, we had both, gDNA and scDNA. We additionally obtained bulk DNA samples for GM7224 and GM7225 that are parents of GM7228. Multiple samples (clones) were processed per cell line (Figure 3.1).

3.2.1 Groundtruth genotype from gDNA

Great amount of supporting data allowed us to create high confidence genotype for both, GM7228 and GM12878 (Figure 3.1). We used the parental information for GM7228 and consensus approach to retain only genotypes that matched the parental inheritance pattern and were 100% concordand within the samples. For GM12878, only retained SNPs that matched the deep sequencing study from Eberle et al. (2017) and were 100% concordant within the samples (Figure 3.1)

3.3 Comparison of scDNA to gDNA

We compared the scDNA genotypes of GM12878 and GM7228 with their corresponding reference created from gDNA (Section 3.2.1). We used two genotyping strategies:

- Minimal quality check (QC001) – genotypes with minimal GenCall threshold accepts virtually all calls

- Standard quality check (QC015) . genotypes with standard GenCall threshold recommended by the vendor

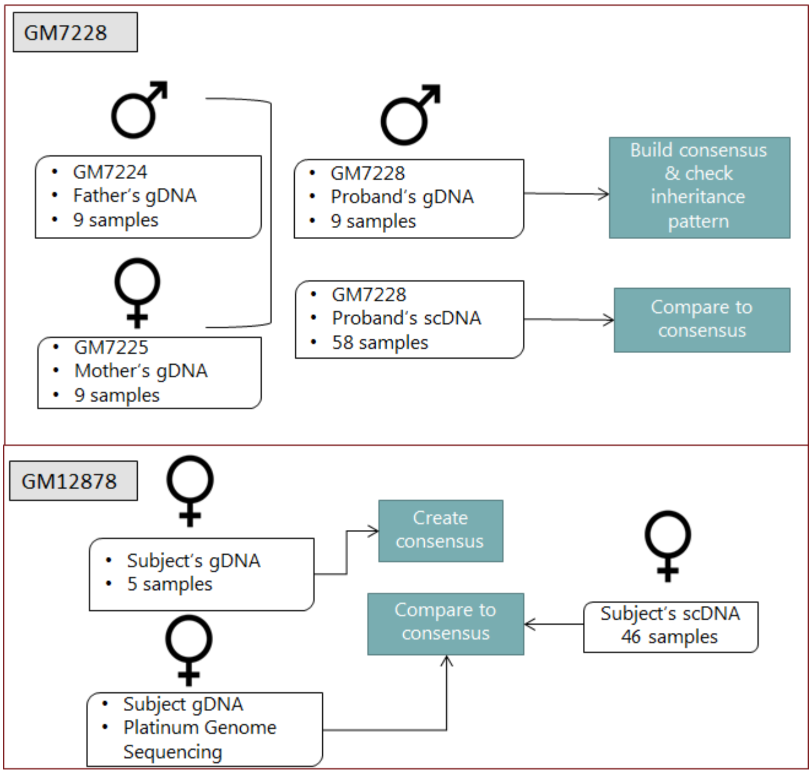


Figure 3.1. Strategy for creating high confidence groundtruth genotype and comparison to scDNA to create reference population of single cell data. For the individual GM7228 there were parental genomes available (GM7224 and GM7225). For the individual GM12878, there wre results of deep next-generation sequencing analysis (Platinum Genome Sequencing, Eberle et al., 2017) available.

Table 3.1 Rules for evaluating the correct genotypes based on parental information

mother (GM07224)	father (GM07225)	proband (GM07228)
AB	AB	{AB,AA,BB}
AA	AA	{AA}
BB	BB	{BB}
AB	BB	{AB,BB}
AB	AA	{AA,AB}
AA	BB	{AB}

Table 3.2. Summary of genotype calls from single cells[a]

	QC001[b]						QC015[c]					
Cell line[d]	+		-		NC		+		-		NC	
	M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
GM 07228	0.39	0.02	0.05	0.01	0.05	0.01	0.36	0.02	0.04	0.01	0.1	0.01
GM 12878	0.4	0.02	0.06	0.02	0.04	0.01	0.37	0.03	0.04	0.01	0.09	0.02
Total	**0.8**		**0.11**		**0.09**		**0.73**		**0.08**		**0.19**	
Total counts	**28.7 million SNPs, 104 cells**											

[a] values are proportions; [b] GenCall cutoff 0.01; ; [c] GenCall cutoff 0.15; [d] +: concordant with reference genotype; -: disconcordant with reference genotype; NC: no calls; M: mean; SD: standard deviation

Using the standard QC015, 73% SNPs from the two single cell datasets called correctly, whereas 8% SNPs were not concordant with the reference genotype (Table 3.2). 19% SNPs gave `no calls` The true positive rate was higher when we used a minimal QC (0.01) compared to the standard QC of GenCall (39%, SD=0.02% and 36%, SD=0.02%, respectively, for cell line GM07228 and 40%, SD=0.02% to 37%, SD=0.03% for GM12878, Table 3.2). In total for both datasets, the GenCall algorithm in its standard configuration (QC015) rejects about 7% of correctly genotyped SNPs from WGA DNA.

3.4 Structure of noise in single cell

Systematic comparison of the scDNA with the reference reveals presence of noise (Table 3.2). To test whether the noise creates a distinct pattern, we performed following steps:

1. We randomly selected 10,000 SNPs from the single cell data GM07228 and extracted the normalized itensities.
2. We transformed the intensities using logarithmic transformation (MA transformation; Vogel, 2019)
3. We estimated density function separately for erroneous calls and correct calls on the M and A values using bivariate normal kernel (Genton, 2002).

The results (Figure 3.2) indicate that, as expected, the correctly genotyped SNPs (positive class) build three clusters corresponding to AA, BB and AB genotypes. The noisy data (negative class) also builds three clusters corresponding to the transition between AB and AA or AB and BB (two clusters of allele drop-outs; ADO) and onec luster with allele drop-ins (ADI). The data suggests good separability of the errorneous clusters from the correct clusters since the centers of the clusters are non-overlapping.

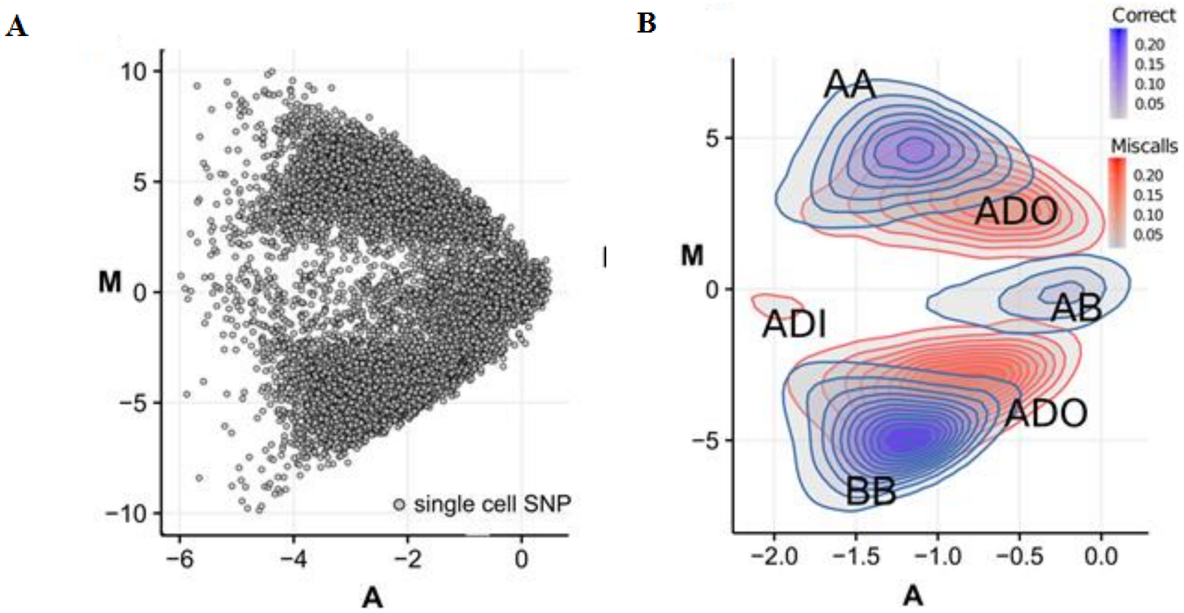


Figure 3.2. MA plot of 10,000 randomly selected SNPs from single cell data. Every dot corresponds to one variant from single cell (A). 2D density function with bivariate normal kernel was applied on the data to reveal the erroneous clusters (B). Red clusters correspond to mistyped SNPs (noise), blue clusters show true signal. ADO clusters mark erroneous transitions from heterozygous to homozygous genotype and ADI cluster marks erroneous transition from homozygous to heterozygous genotype.

3.5 Training dataset

3.5.1 Feature selection

We compared our single cell datasets to the reference genotype in Section 3.3. More specifically, for every candidate single cell call for SNP i and sample s we assigned a label $l_{i,s}$: $l_{i,s} \in \{True, False\}$, depending on the match or mismatch with the corresponding reference genotype call. The training dataset is then a set of of triplets $(m_{i,s}, a_{i,s}, l_{i,s})$, where $(m_{i,s}, a_{i,s})$ are input features and $l_{i,s}$ is the output feature. Note that we omit sample index s in further explanation, as we do not distinguish between the origins of SNPs in the training data set. We included all autosomal single cell calls with GenCall score above 0.01 (QC001) totaling 14,403,139 SNPs for training (GM07228) and 11,737,508 SNPs for validation (GM12878). Lowering the GenCall score threshold for accepting a SNP allowed us to include potentially poorly amplified SNPs and to capture the full error pattern.

3.5.2 Problem of imbalanced dataset

Table 3.2 suggests that the training data is highly imbalanced with positive class being the majority class. The positive class is our target class, and the analysed single cell datasets reflect the `real world` class ratio. Neverthless, as we operated with sufficient amount of data we performed downsampling of the positive datasets to obtain the same amount of positive and negative samples.

3.6 SureTypeSC

SureTypeSC is a novel machine learning method that combines nonparametric (in terms of statistical distribution) supervised method embodied in Random Forest (RF) that adapts to the noise in the single cell. RF is trained on the reference dataset described in Section 3.2. The fitted RF then estimates the regions of noise and high quality SNPs in unseen data. These regions are then formalized using a parametric model. This is a second layer of the algorithm and consists of a system of Gaussian mixtures called Gaussian discriminant analysis.

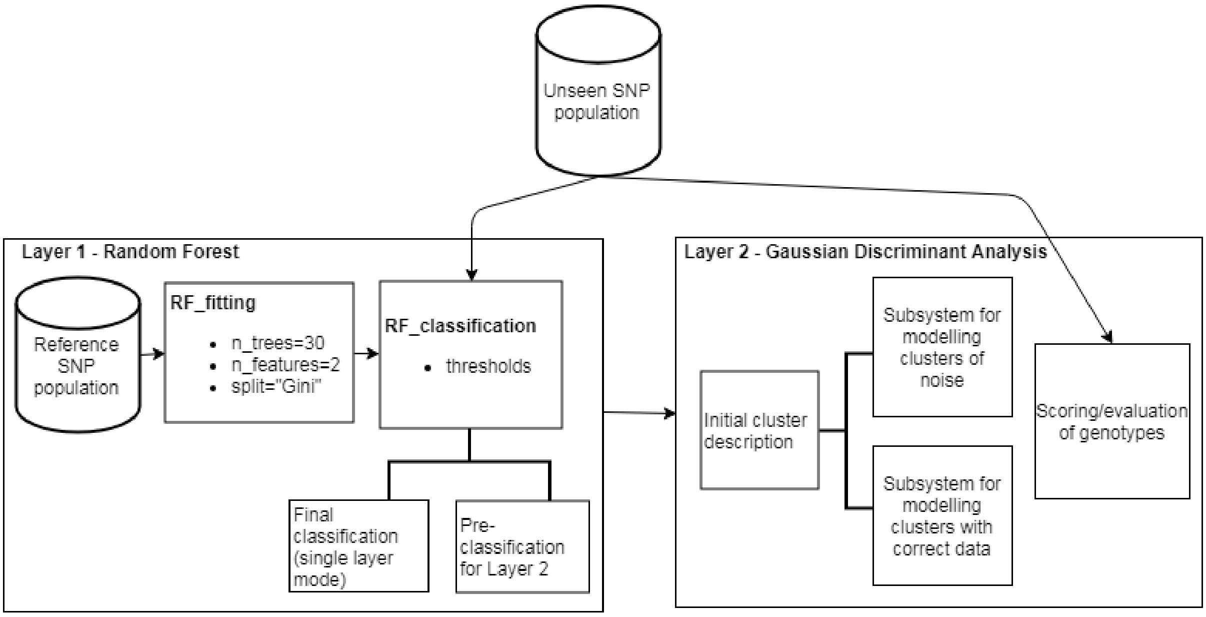

Unseen SNP population
Layer 1 - Random Forest
Reference SNP population
RF_fitting
n_trees=30
n_features=2
split="Gini"
RF_classification
thresholds
Final classification (single layer mode)
Pre-classification for Layer 2
Layer 2 - Gaussian Discriminant Analysis
Initial cluster description
Subsystem for modelling clusters of noise
Subsystem for modelling clusters with correct data
Scoring/evaluation of genotypes

Figure 3.3. Scheme of a prototype for SureTypeSC based on machine learning with two layers organized in cascade.

The prototype of SureTypeSC is schemed in Figure 3.3. Classification of unsen SNP data is first performed with Layer 1 (RF trained on reference data). RF in the single layer mode directly classifies the unseen data. In the standard, two layered mode, RF estimates the positive and negative classes for the further improvement with Layer 2. Layer 2 accepts the classification from Layer 1 and creates an initial cluster estimate, which is then iteratively refined. Subsequently, the genotypes from the unseen SNP population are evaluated again with the fitted model from Layer 2. Figure 3.3 also illustrates parameters of the model, that can be subsequently subjected to experiments. This is i.e. threshold for the positive class defined for RF. This parameter influences the input of the Layer 2 and therefore the output of the final classification. Experiments with this and other parameters of the model are described and discussed in Vogel (2019). The following two subsections discuss both layers of the algorithm in a greater detail.

3.6.1 Basic Random Forest layer

We chose RF (Breiman, 2001) for the initial training and classification. The kernel density estimation in Section 3.4 suggests that the function that separates the erroneous clusters from the clusters of correct data (red and blue contours in Figure 3.2B) is non-linear. The ensemble nature of Random Forest has by definition the ability to fit different trees to different parts of the input space and therefore mimic a non-linear separating function that, in theory, can increase the classification accuracy.

3.6.2 Refinement using Gaussian Discriminant Analysis

The second layer of the algorithm is Gaussian Discriminant Analysis (GDA) that formalizes the genotype clusters obtained from the RF step and potentially improves the classification. GDA models positive and negative class separately using GMM and defines a scoring function that discriminates the data based on their affinity to positive and negative class. The general concept of GDA was adapted from Ng (2019) and Hastie et al. (2016).

Let $D = \{x_j | j = 1 \dots N, x_j \in \mathbb{R} \times \mathbb{R} \times \mathbb{G} \times \widehat{\mathbb{L}}\}$ denote a set of N SNPs that were classified by the trained RF, where $\mathbb{G} = \{\mathrm{AA}, \mathrm{AB}, \mathrm{BB}\}$, $\widehat{\mathbb{L}} = \{\mathrm{T}, \mathrm{F}\}$. Therefore, $x_j = (m_j, a_j, g_j, \widehat{l_j})$ is a quadruplet of the logarithmic difference, logarithmic average, genotype predicted by GenCall (QC 0.01) and class prediction by RF at the *j*-th SNP. We assume that both the positive (T) and negative (F) classes, which are represented by pairs $d_i = (m_i, a_i)$, are drawn from mixtures of multivariate normal distributions and define the following system of Gaussian discriminants:

$$\hat{L} \sim Bernoulli(\lambda) \tag{3.1}$$

$$p(\hat{l}) = \lambda^{\hat{l}} (1 - \lambda)^{1 - \hat{l}} \tag{3.2}$$

$$p(d_j | \hat{l} = T) = p(d_j | \Theta_{\mathrm{T}}) = \sum_{k=1}^{3} \alpha_{T,k} \phi(d_j | z_{T,k}, \theta_{T,k}) \tag{3.3}$$

$$p(d_j | \hat{l} = F) = p(d_j | \Theta_{\mathrm{F}}) = \sum_{k=1}^{3} \alpha_{F,k} \phi(d_j | z_{F,k}, \theta_{F,k}) \tag{3.4}$$

Where:

- λ denotes probability $p(\hat{l} = T | d_j)$
- ϕ is multivariate normal density function with parameters θ_k (with mean μ_k and covariance matrix Σ_{k})
- z_k is an indicator variable that denotes the genotype class, where $z_k \in \mathbb{G} \times \widehat{\mathbb{L}}$
- 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 $\Theta_{\widehat{\mathbb{L}} \in \{T,F\}} = \{\alpha_{\widehat{\mathbb{L}},1}, \dots \alpha_{\widehat{\mathbb{L}},3}, \theta_{\widehat{\mathbb{L}},1} \dots \theta_{\widehat{\mathbb{L}},3}\}$.

The log-likelihood function $\mathcal{F}$ for classes from $\widehat{\mathbb{L}}$ is defined as follows:

$$\ln \mathcal{F}(\Theta) = \sum_{j=1}^{N} \ln p(d_j | \Theta_{\hat{L}}) \tag{3.5}$$

We use an Expectation Maximization algorithm (Dempster et al., 1977) to estimate the parameters Θ_L of the positive and negative class that maximize their log-likelihood function (Eq. 3.5). The EM algorithm is divided into an Expectation-Step (E-Step) and a Maximization-Step (M-Step). These are run in iterations separately for the positive and negative classes until convergence is reached. $N_{\hat{l}}$ is total number of SNPs in the particular class. The details of the algorithms are in Vogel (2019).

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 i is evaluated for both Gaussian discriminants using the following formula:

$$\left(score_{i,T}, score_{i,F}\right) = [\ln p(d_i | \Theta_{\hat{T}}), \ln p(d_i | \Theta_{\hat{F}})] \tag{3.6}$$

The final classification (membership to a positive or a negative class) is determined by higher value from the pair $(\mathrm{score}_{\mathrm{i,T}}, score_{i,F})$.

3.6.3 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 GenCall having its own scoring scheme, we used the following equations to estimate the probability of a certain SNP being correctly genotyped:

1. Random Forest: the score of a genotype of the *i*th SNP is given as a proportion of the trees in the forest that voted for a particular genotype being correct:

$$score_{i,RF} = p(l_i = T | d_i) \tag{3.7}$$

2. The scoring strategy of SureTypeSC is inferred from its second layer (GDA) as the class-conditional posterior probability of a genotype falling into positive class T:

$$score_{i,RF-GDA} = \frac{e^{score_T} \times p(T)}{\sum_{Z \in \{T,F\}} e^{score_Z} \times p(Z)} \tag{3.8}$$

4 Validation

The validation strategy for the novel model for filtering the single-cell data presented in Chapter 3 is summarized in Figure 4.1. The results of the validation part were published in Vogel et al (2019).

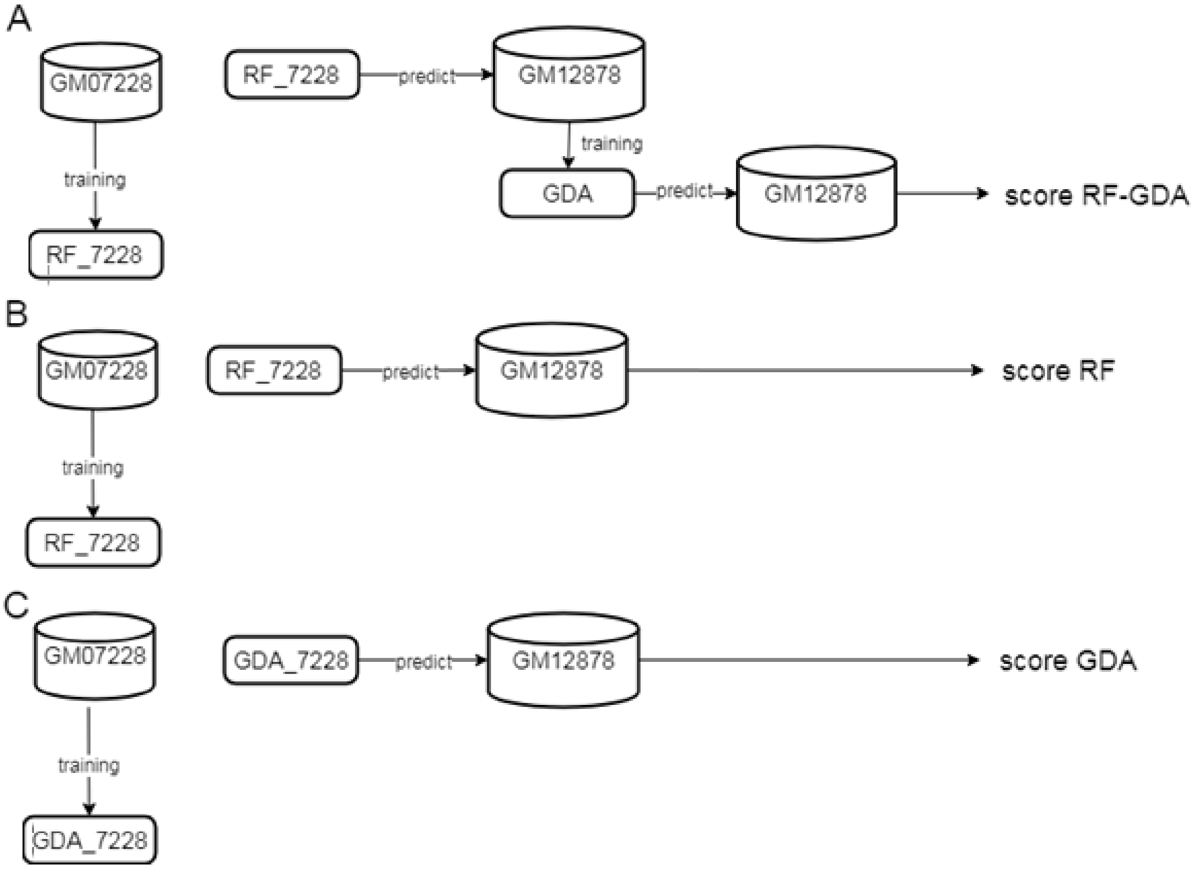


Figure 4.1. Training and testing strategy for SuretypeSC. (A) The RF was trained on ground truth data from GM07228 and used to predict the values of GM12878. The GDA was used to fit the predicted values of GM12878 (B) The RF was trained on the ground truth data from GM07228 and used for prediction and scoring on the testing data, GM12878. (C) The GDA trained on ground truth from GM07228 and prediction and scoring took place on the testing data, GM12878.

We analyse the performance of SureTypeSC, but also its constitutive single layers. We therefore denote, consistently with the previous explanations, the single layers by their acronyms (RF and GDA) and the cascade solution by their combination (RF-GDA). To pinpoint the differences in performance and to show the improvements SureTypeSC achieves in the single cell domain, we included the performance of GenCall in all validation analyses. GenCall represents the current state of the art and has been used in multiple single-cell genotyping analyses (Ottolini et al., 2016; Esteki et al., 2015; Handyside et al., 2010). As we operate with multiple single-cells (46 or 58 cells; Section 3.2), we calculate mean values and standard deviation or confidence intervals to capture the variability. We refer the reader to Vogel (2019) for test of statistical significance of the differences between the performances of the algorithms. We always evaluate the heterozygous and homozygous genotypes separately. Having reliable method for heterozygous loci improves detection power of many knowledge extraction algorithms from single cell data (Chapter 5 and Chapter 6). We use validation metrics from two categories:

- procedures that capture the whole spectrum of classification outcomes (visual representation and ROC-AUC score);
- validation metrics that operate with fixed classification thresholds; these procedures mimic the real applications where we have to decide for a particular cutoff to retrieve the information about rejecting or accepting a particular genotype; we used accuracy, precision, recall and F1-score

We discuss the antagonistic relationship of precision and recall and demonstrate different threshold settings to allow high recall or precision of the algorithm.

4.1 Validation curves and ROC-AUC score

We first demonstrate the performance of the algorithms using ROC and Precision Recall curves. These metrics gave us visual insight into overall performance of the classifiers, invariant to the score cutoffs used. For the heterozygous calls, RF-GDA outperforms all tested algorithms (Figure 4.2, Figure 4.3 and Table 4.1). While GenCall achieves a 74% ROC-AUC score on average, this is increased to 86%, 87% and 92% for RF, GDA and RF-GDA, respectively (Table 4.1). For the homozygous regions, the RF outperforms GenCall at all points of the ROC and Precision-Recall curves, which is supported by the increase in the ROC-AUC score from an average of 67% (GenCall) to 81% for the RF (Table 4.1 and Figure 4.2A). This is further increased with the GDA or RF-GDA (both 83%, Table 4.1). Interestingly, at a precision of approx. 93%, the RF curve crosses that of the GDA and RF-GDA and recalls more true positive homozygous calls (Figure 4.3). This suggests that the RF alone might be a good option if higher recall is required at the costs of lower precision, which is nevertheless higher than GenCall in the homozygous regions. GenCall crosses the Precision-Recall curve of the RF-GDA at a precision around 88% and recalls more true positives (Figure 4.3A). This is, however, very close to a recall of 100%, which also means accepting all calls without any filtration.

Table 4.1 ROC-AUC score of the genotyping algorithms on independent dataset GM12878[a]

GenCall		**RF**		**GDA**		**RF-GDA**	
het	homo	het	homo	het	homo	het	homo
0.74±0.01	0.67±0.015	0.86 ± 0.004	0.81 ± 0.012	0.87 ± 0.005	0.83 ± 0.013	0.92 ± 0.004	0.83 ± 0.012

[a] values are mean proportions over 46 cells ± confidence interval at 95%

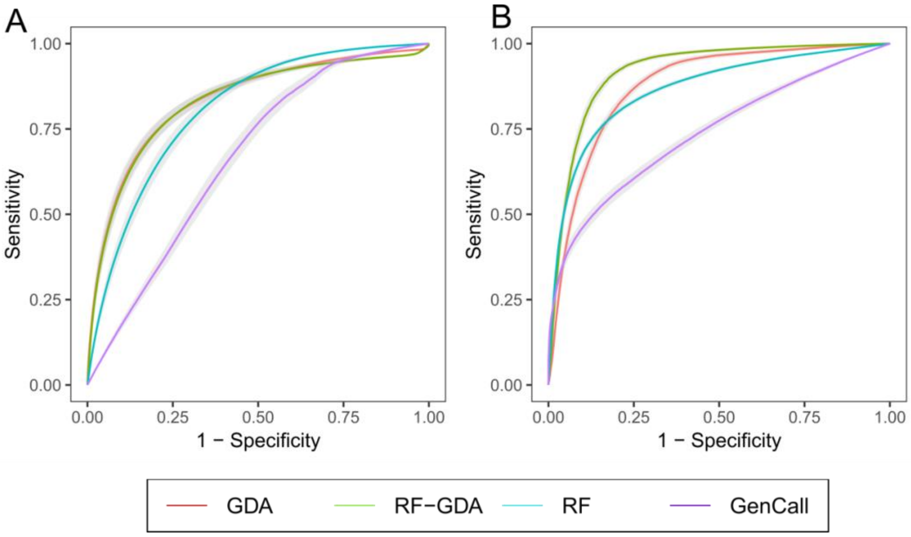


Figure 4.2. ROC curve for homozygous (A) and heterozygous (B) calls.

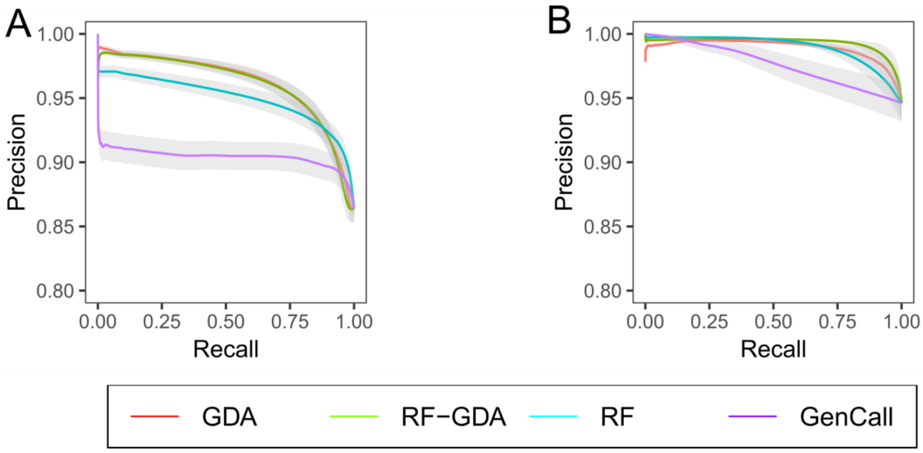


Figure 4.3. Precision-recall curve for homozygous (A) and heterozygous (B)

4.2 Evaluation with fixed thresholds

In this evaluation strategy, we selected recommended classification threshold for the GenCall algorithm. For RF, GDA and RF-GDA, we selected thresholds that emphasize on the differences in performance Results show that in this configuration, GenCall recalls 68% of the true positive heterozygous genotypes at precision of 97% (Table 4.2). The RF-GDA has 84% recall and achieves average precision of 99% and thus outperforms GenCall in both precision and recall. Having similar precision than RF-GDA, single layers RF and GDA recall fewer true positive heterozygous genotypes (Table 4.2).

High precision and recall are reflected in high harmonic mean of precision and recall (F1-score) for the RF-GDA (Table 4.2) and high rate of correctly classified SNPs (accuracy, Table 4.2). GenCall recalls 96% of the true positive homozygous genotypes on average at precision 89%. At similar recall, the RF alone increases precision by 2%. GDA and RF-GDA further improve precision, but at the cost of recall. Both methods achieve an average precision of 92% at 90% recall for the homozygous calls (Table 4.2). Recalling fewer true positives at higher precision causes a drop in the F1-score for GDA and RF-GDA. This is because recall declines much quicker than the precision increases (Figure 4.3A). The effect of lower recall from the GDA and RF-GDA is also mirrored in the lower accuracy. As GDA and RF-GDA have higher precision, they are also more likely to reject correct SNPs, thereby decreasing the number of true positives.

Table 4.2 Performance of the genotyping algorithms on independent dataset GM12878[a]

Alg. **Metrics**	**GenCall[b]**		**RF**		**GDA**		**RF-GDA[g]**	
	het	homo	het[c]	homo[d]	het[e]	homo[f]	het	homo
accuracy	0.68 ± 0.01	0.86 ± 0.012	0.71 ± 0.013	0.88 ± 0.008	0.63 ± 0.009	0.85 ± 0.01	0.84 ± 0.014	0.85 ± 0.01
F1-score	0.8 ± 0.01	0.92 ± 0.007	0.82 ± 0.012	0.93 ± 0.005	0.76 ± 0.011	0.91 ± 0.007	0.91 ± 0.011	0.91 ± 0.007
precision	0.97 ± 0.01	0.89 ± 0.009	0.99 ± 0.001	0.91 ± 0.008	0.99 ± 0.001	0.92 ± 0.008	0.99 ± 0.001	0.92 ± 0.008
recall	0.68 ± 0.01	0.96 ± 0.005	0.7 ± 0.017	0.96 ± 0.001	0.61 ± 0.013	0.9 ± 0.005	0.84 ± 0.017	0.9 ± 0.006

[a] values are mean proportions over 46 cells ± confidence interval at 95%; [b] GenCall score threshold 0.15; [c] Random Forest score threshold 0.6 and [d] 0.15 ; [e] Gaussian Discriminant Analysis score threshold 0.8 and [f] 0.5 ; [g] RF-GDA score threshold 0.15

5 Knowledge extraction from single cell data

As presented in the generic workflow in Figure 2.1, creating a model over the intensity data and genotyping is not a final step of the analysis. The whole workflow would have zero practical impact without proper interpretation of the data and analysis of the context. We call this process knowledge extraction and will present few algorithms from this category. The presented algorithms are either optimization of the previously published concepts or present original solutions for knowledge extraction – this is clearly distinguished in the text.

5.1 Data model of recombination in meiosis

5.1.1 Introduction

Meiosis is a type of cell division that gives rise to human reproductive cells. An important mechanism that occurs during development of reproductive cells is homologous recombination. It is a process, where the parental homologous chromosomes swap genetic material (swap of greeb and yellow in Figure 5.1B).

The genetic information inherited from the same parent (here, continuous stretch of green or yellow in Figure 5.1) is called haplotype block. During the first stage of meiotic divison, the homologous chromosomes segregate and give rise to two different cells – oocyte[4] and first polar body (PB1, Figure 8.1C). The sister chromatids segregate upon fertilization or artificial activation of oocyte in the lab (Models II-1 and II-2, respectively, Figure 5.1D,E; Ottolini et al., 2015). During this process, second polar body (PB2) is extruded (Figure 5.1D,E).

5.1.2 Meiotic pathways and data models

Thanks to advanced laboratory technologies, we are able to obtain data for most of the pathways (models) depicted in Figure 5.1 using model in Figure 2.1 and SNP arrays (Section 2.1.2). Analogously to Chapter 3, we can identify two categories of data – accurate information from bulk genomic DNA (Figure 5.1A) and noisy genotype information from single cell DNA represented by the products of female meiosis (Figure 5.1C,D,E).

---

[4] we use the term oocyte and egg interchangeably

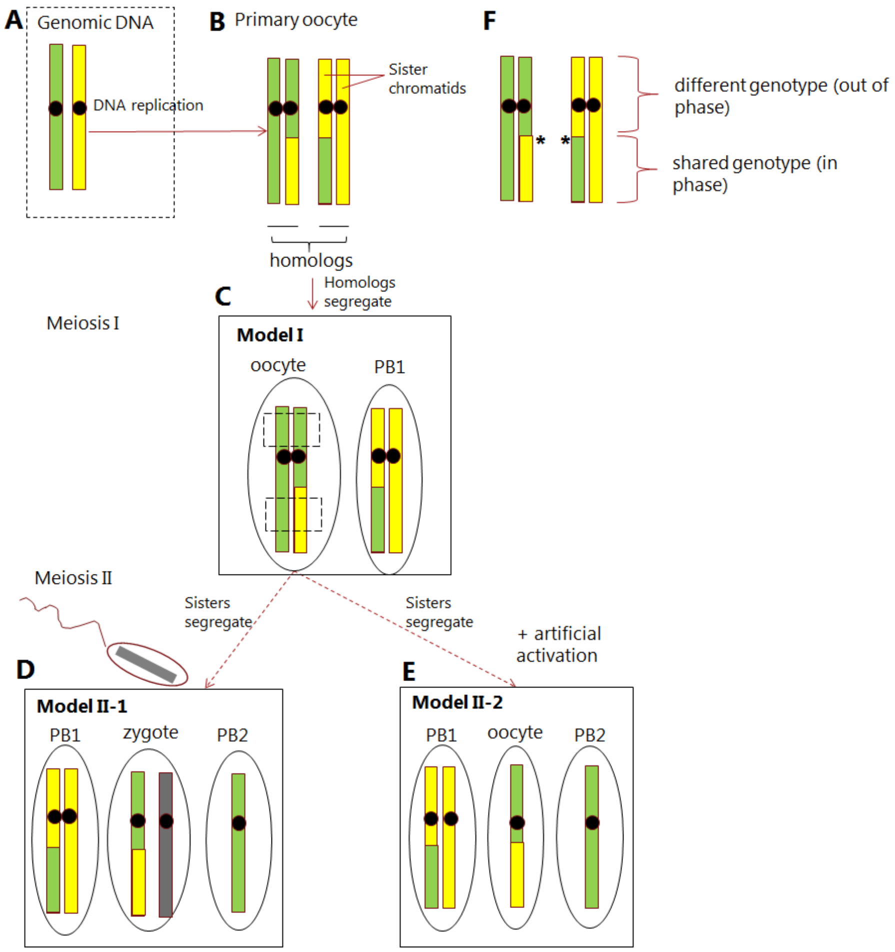


Figure 5.1. Generation of human oocytes. The genomic DNA (A) is replicated and undergoes recombination (B). The homologous chromosomes segregate and give rise to two different cells – oocyte and polar body (C). The sister chromatids segregate upon fertilization (D) or artificial activation of oocyte in the lab (E).(F) Illustration of crossover point. After recombination, the two diploid cells share their recombinant DNA.

5.1.3 Model I – reconstructing crossovers from Duos

Figure 5.1C shows stage of the meiotic division with two cells that both posses two strands of DNA. It is apparent that the recombinant parts of the chromosome share the same combinations of alleles (green and yellow building heterozygous genotype), whereas the part of the chromosome that did not recombine contain only information from one parent (Figure 5.1F). If we therefore compare the genotypes from these two cells and mark matching genotypes with 1 and varying genotypes with 0, we would ideally obtain a regular expression $[0]\{n-m\}[1]\{m\}$, where n is the size of the chromosome and m is the size of the recombinant part of the chromosome. The transition point between 1 and 0 would represent the crossover. However, following needs to be taken into accound: (a) Genotypes will also match by chance and (b) Genotypes contain errors causes by WGA. The observed sequence at the output is therefore $[0|1]\{n\}$, where n is the size of the chromosome.

Design of a new algorithm

Let AB→ ~0.5, AA→ ~0 and BB→ ~1, where the (real) numbers are B-allele frequences inferred from the normalization procedure of the GenCall algorithm (Vogel, 2019). The algorithm assumes that the differences between the regions that did not recombine are high and iteratively marks the absolute distances of the B-allele frequences of the loci of the two cells.. It is however assumed that these regions can also contain markers with zeros when the corresponding homozygous SNPs from two cells match by chance as explained previously. Sufficiently high threshold (currently 0.95), however, filters these markers out (function *MarkAndSelect(.)*)

The candidate homozygous loci (with markers close to 1) then undergo a segmentation procedure. (SegmentHomRegions(.)), that performs segmentation on the homozygous regions and separates them into potentially multiple homozygous segments. For the actual segmentation, a Variational Bayesian GMM is employed, as this allocates number of segments dynamically (Blei and Jordan, 2006). The boundaries of these segments then determine the positions of the crossovers. The pseudocode of the algorithm is in Vogel (2019).

5.1.4 Model II – reconstructing crossovers from Trios

The algorithm for finding crossovers in Trios is more complex due to number and types of cells involved – we operate with diploid PB1 and haploid PB2 and Oocyte (Section 5.1.2). It firstly only selects heterozygous SNPs from maternal genomic DNA (Figure 5.1A). These are termed as informative, as they contain both alleles that can be tracked in the PB1, PB2 and Egg (Figure 5.1D). Secondly, the algorithm compares one of the haploid cells (PB2 or Oocyte) from the same individual to all the other cells. This haploid cell is termed as hypothetical common ancestor or reference and its purpose is to determine the origin of the haplotype blocks (Figure 5.1). We call the regions that are shared with the reference as `in phase` and the procedure of determining the origin of haplotype blocks as phasing. The transition between two haplotype blocks is crossover. However, the reference, although initially assumed as a homogenious haplotype block, also potentially carries crossovers. If this is the case, an artificial crossover would appear in all the other cells that underwent phasing .

The original implementation of the algorithm requires user interaction and manipulation throughout the analysis (Ottolini et al., 2016). I.e., the transitions between the haplotype blocks and common crossovers need to be visually inspected and marked and corrected manually. We reflected these drawbacks on following changes to the design of the algorithm:

- we automated the process by designing algorithm for resolving common crossovers and transitions between the haplotypes
- we optimized the algorithm by designing operations on matrices and vectors that can be efficiently implemented in one of the high level numerical libraries (Numpy for Python, Walt et al., 2011)

The pseudocode of the optimized algorithm is shown in Algorithm 1. Note that the requirements for minimal number of input trios ($n >= 3$) were previously validated in Ottolini et al. (2016).

Algorithm 1: Crossover detection in Trios

```
Input: D={n vectors of genotypes- gDNA, PB1, PB2 and Egg} from the same individual}, where |n|=1+3x and x is number of trios (>=3)
Output: Cx={list of crossover positions per cell per chromosome}
    1. Mask out loci where $gDNA \in \{AA, BB\}$ and $PB2 = AB$ and $Egg = AB$
       //we only operate with informative SNPs and het PB2 and Egg are likely an error
    2. Convert genotypes to B-allele frequencies (AA←0, BB←1 AB←0.5)
    3. Phase(D, Ref): perform | X-Ref |, where $X \in \{PB1, PB2, Egg\}$
        a. Return $D_{phased}$
    4. Smooth_Haplotype($D_{phased}$): apply 1D mode filter to phased data
        a. Return $D_{smoothed}$
    5. ResolveCx($D_{smoothed}$) – resolve common crossovers in reference
    6. Phase($D_{smoothed}$, $Ref_{resolved}$)
    7. Report crossovers and haplotype blocks
```

Function Phase

The function applies fast arithmetic operations to all chromosomes in all cells in the trio that compares all data to the reference cell. It returns mask to every chromosome present in the dataset with 0 for loci in phase with the reference, 1 for loci out of phase and 0.5 for heterozygous loci.

Function Smooth_Haplotype

Function processes the phased genotypes with 1D mode filter. 1D mode filter is analogous to median filter with mode as the smoothing function. Median filter is a non-linear technique to remove noise (or generally outliers) from the data and prevent edges (Bovik et al., 1987). The edges are boundaries of the phases/haplotypes (and therefore crossover points) and there are two categories of outliers present in the data:

- True technical noise caused by whole genome amplification
- Heterogeneity of the heterozygous region as this can not only possess AB, but also AA and BB if the alleles are matching by chance (Section 5.1.3)

The task is to unify the phases by removing the erroneously phased genotypes caused by aforementioned reasons, but preserve the edges (crossovers) to obtain two haplotypes and a heterozygous region. The details and the pseudocode of the algorithm is in Vogel (2019).

Function ResolveCx

Function creates a matrix of crossovers and subsequently searches for common crossovers that indicate crossover in the reference cell. As crossovers are defined by transition of haplotypes in a matrix (haplotype transition is indicated with 1 in matrix *cx*, Algorithm 2), the number of common crossovers per locus is sum over all columns (*colSum* in Algorithm 2) The crossover is then subsequently resolved and phases are added to the reference sequence.

Algorithm 2: Function ResolveCx

Input: *Collection of phased and smoothed trios D_{phased} from the same individual including reference Ref without crossovers*

Output: *$Ref_{resolved}$ with resolved crossovers and added phases*

```
1. Init matrix of crossovers cx of the same  dimensions as  $D_{phased}$
2. For cell in $D_{phased}$::
    a. For pos in cell
        i. cx[pos,cellid] ← phase[i]!=phase[i+1]
    b. For row in cx:
        i. If colSum(row) = total number of haploid cells in $D_{phased}$
            a. Ref[row] ← 1
3. Init $Ref_{resolved}$ with dimensions identical to Ref
4. For pos in Ref:
    a. $Ref_{resolved}$ [pos] = cumSum(Ref[pos]!=Ref[pos+1]) mod 2
```

Note that step 2ai in Algorithm 2 marks all transitions between haplotypes and then 2bi checks for common crossovers. Once the crossovers are added to the reference, step 4a corrects for haplotype transitions between introduced crossovers using cumulative sum (*cumSum*).

After the common crossovers are resolved, the cells undergo a second round of phasing – now with a phased reference. The phasing procedure (Algorithm 2) is universal in this matter and can be again used to adjust the phases of all cells in respect of crossover in the reference cell.

5.2 Gene conversions

5.2.1 Introduction

Crossover and recombination contributes to heterogeneity of the human population by shuffling the genetic information during meiosis. While crossover is a reciprocal transfer of genetic information that maintains the same amount of alleles for every locus, so called gene conversions are non-reciprocal. A required condition for both, crossover and gene conversion is presence of double strand breaks (DSBs). DSBs are subsequently resolved by either crossover, or gene conversion, where the missing gap in DNA is synthetised in favor of one of the alleles. The detailed mechanism is explained thoroughly elsewhere (Chen et al., 2007).

5.2.2 Detection of GCs in Trios

The detection of gene conversions is based on mendelian inheritance patterns (law of segregation). The design of the algorithm is summarized in Algorithm 3 and the principle is straightforward – every locus that shows allelic imbalance (different counts of allele A and B over all three cells per locus) is a potential gene conversion.

```
Algorithm 3: Detect GC
Input: quadruplet of genotype vectors Q={gDNA, PB1, PB2, Egg} from a single individual
Output: list of GC
    1. Convert genotypes to B-allele freq: AB ← 0.5, AA ← 0, BB ← 1.0
    2. Select loci where gDNA=0.5 and PB1=0.5
    3. For every locus l in Q:
        a. If $4 \times gDNA[l] - (2 \times PB1[l] + PB2[l]) \; != 0$
            i. Report GC at locus l
```

As incidence of ADI is much lower than ADO, and the probability of transition between two homozygous genotypes (AA ↔ BB) is low, Algorithm 3 only takes loci where PB1 is heterozygous. Due to potential noise present in the data, Algorithm 3 in this form requires high confidence genotypes (application to real data in Chapter 6).

5.3 Conclusion

In this chapter we presented algorithms for knowledge extraction from single cell data, namely data from female meiosis. We show applications of these algorithms on real data and dicsuss how noise affects the quality of the knowledge extraction in the next chapter.

6 Applications of SureTypeSC

6.1 Introduction

In this chapter, we demonstrate practical applications of the novel algorithm for filtering genotypes (Chapter 3) and algorithm for knowledge extraction from single cell data (Chapter 5). We will use single cell data from products of female meioses (Chapter 5) or reference single cell data (Chapter 3). The goal is to show performance of the novel filtering algorithm on these data in comparison with the state of the art algorithm GenCall and discuss the quality of genotyping in relation to the knowledge extraction.

6.2 Improved crossover detection

6.2.1 Duos

We demonstrate the functionality of the algorithm for crossover detection in Duos (Section 5.1.3) on chromosome 1 of a PB1-oocyte duo from Gruhn et al. (in resubmission). We were interested how the algorithm performs on original unfiltered data and with different stringencies of GenCall (QC015, QC087 and QC095) and SureTypeSC (RF-GDA015, RF-GDA057, RF-GDA075).

The results suggest, that increased stringency contributes to clearer homozygous segments by removing poor quality signal – this is true for both, GenCall and RF-GDA. (Figure 6.1) However, RF-GDA shows higher specificity towards the noise for similar number of markers (QC087 and RF-GDA have 8,799 and 8,751 points in Figure 6.1D and Figure 6.1E, respectively. This is also supported by highly stringent GenCall (QC095), that still fails to reject few likely false positives (dispersed green points around green cluster in Figure 6.1F).

RF-GDA075 effectively cleans up the signal and enables clearer separation of the homozygous clusters with the segmentation algorithm. Note that the red singleton in Figure 6.1G could be either FP left out by SureTypeSC or gene conversion (Section 5.2). The algorithm currently takes this marker as a boundary for the crossover (Fig. 8.5H).

6.2.2 Trios

We applied for detecting crossovers presented in Section 5.1.4 for detection of crossovers in Trios and were interested in performance of the algorithm with data from standard GenCall (QC015) and data filtered with RF-GDA. We demonstrate the differences between the results on chr17 from a trio from an in-house data collection of Hoffmann Lab at University of Copenhagen[5].

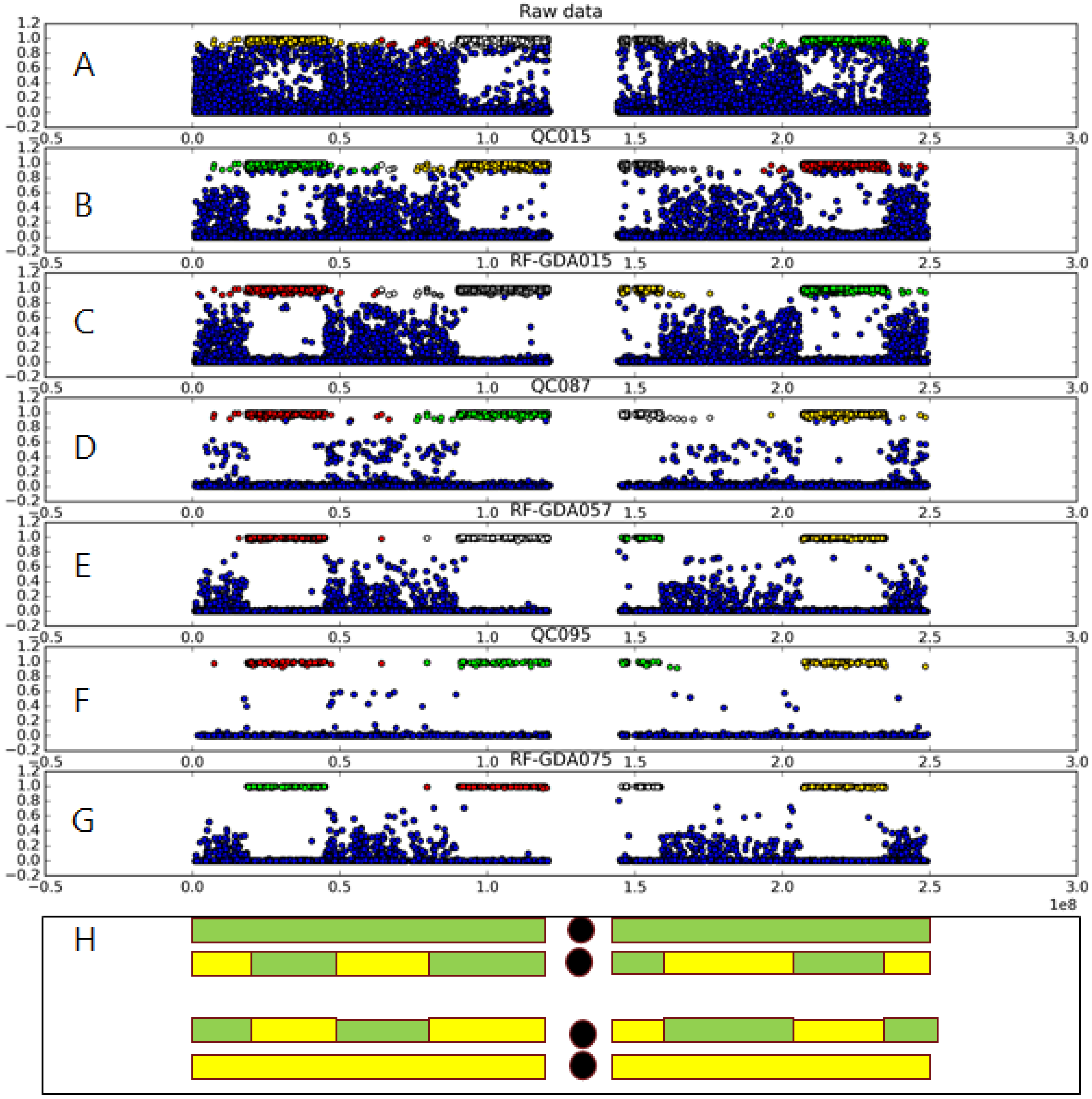


Figure 6.1. Visual representation of the results of the crossover detection algorithm in duos using different filtering strategies. The points other than blue represent clustered segments of homozygous data. (A) Raw data is the genotyping data from GenCall from GenomeStudio with no filtering. (B) is the data fitlered with standard GenCall threshold 0.15. (C) is raw data filtered with algorithm RF-GDA from SureTypeSC using standard threshold 0.15. (D) is GenCall algorithm with stringend threshold 0.87. (E) RF-GDA with threshold that keeps the same amount of markers as QC087. (F) GenCall with very strict threshold close to 1. (G) RF-GDA previously validated for high precision. (H) Likely crossovers inferred from boundaries of the segments clustered from G.

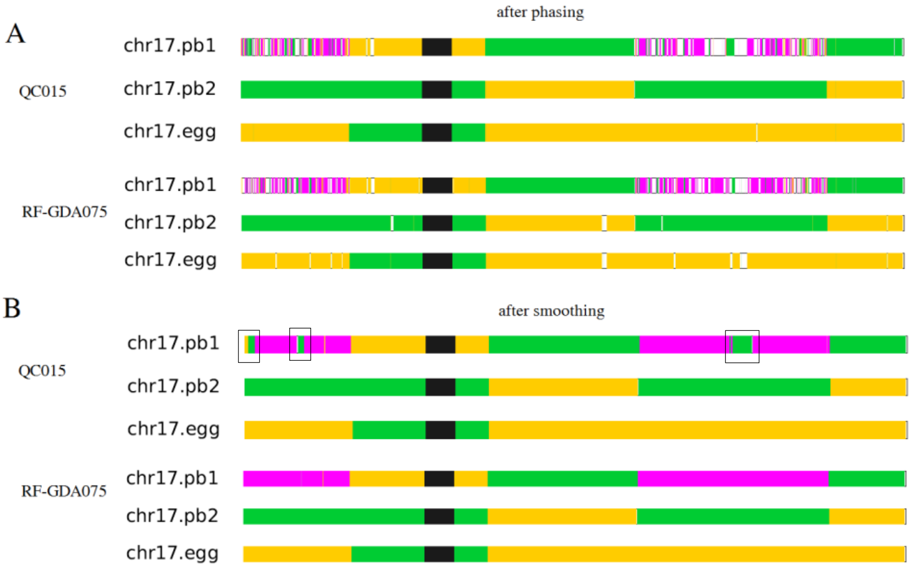
Comparison of the outputs of both genotyping strategies (Figure 6.2) indicates, that:

- The density of the heterozygous SNPs is higher with RF-GDA075 – this is expected and is concordant with the validation study that SureTypeSC can resolve more true positive heterozygous SNPs (Chapter 4).

---

[5] https://icmm.ku.dk/english/research-groups/hoffmann-group/

- As a direct consequence of the previous point, we observe artificial crossovers and transitions between haplotypes in the heterozygous regions after smoothing with data from GenCall (QC015, Figure 6.2). An artificial haplotype is non-reciprocal (lacking support in PB2 and Egg). I.e. haplotype switch in PB1 from heterozygous to haplotype1 should trigger haplotype transition in one of the other cells. These events are marked with rectangles. in Figure 6.2B.



-

Figure 6.2. Results of the the phasing stage and the smoothing stage of the crossover detection algorithm for data coming from GenCall with standard threshold (QC015) and RF-GDA075. Heterozygous regions (containing both green and yellow haplotypes) are in violett , and the parental phases in green and yellow. The solid rectangles indicate artifical haplotype blocks.

6.3 Direct detection of gene conversions

6.3.1 Genotyping with GenCall

Algorithm for detection of gene conversions (Section 5.2) does not distinguish between gene conversions and errors caused by MDA amplification. Running the algorithm on the data with QC015 reveals 2449 gene conversion events with heterozygous PB1 (Figure 6.3). As we are taking into account only heterozygous SNPs that span about 16% in a diploid cell (around 50,000 heterozygous SNPs in 300k microarray). This fraction corresponds to $5.12 \times 10^6 bp$ in human genome. Assuming the gene conversion tract has length around 100 bp on average (Padhukasahasram and Rannala, 2012), 2449 detected gene conversion span roughly 250.000 bp. This corresponds to gene conversion rate of 4.9×10^{-2} per bp, per meiosis. Rate of gene conversion has been previously estimated as 5.9 $\times 10^{-6}$ per bp, per meiosis (Williams et al., 2015), which is roughly 10.000 times less than what we predicted. It is to conclude that most of the detected gene conversions from genotypes using standard genotyping using GenCall false positives.

Table 6.1. Possible combination of alleles where loci in PB1 are heterozygous

PB1	***PB2***	***Egg***	***Status***
AB	*AA*	*BB*	*Mendelian inheritance*
AB	*AA*	*AA*	*Gene conversion BB→AA or ADI in PB1*
AB	*BB*	*AA*	*Mendelian inheritance*
AB	*BB*	*BB*	*Gene conversion AA→BB or ADI in PB1*

6.3.2 Filtering with SureTypeSC

We were futhermore interested in numbers of gene conversion after we filter the genotypes with SureTypeSC aimed for high precision (RF-). Figure 6.3 shows the results on top of the results of standard GenCall. The number of gene conversion events is reduced to only 10. This corresponds to gene conversion rate of 1.9×10^{6} per bp, per meiosis. A lower number than reported in Williams et al. (2015) can be explained by the fact that we only taken into account the heterozygous part of the genome of PB1, but neverthless suggests that we were able to separate noise from the truth signal and detect gene conversions directly.

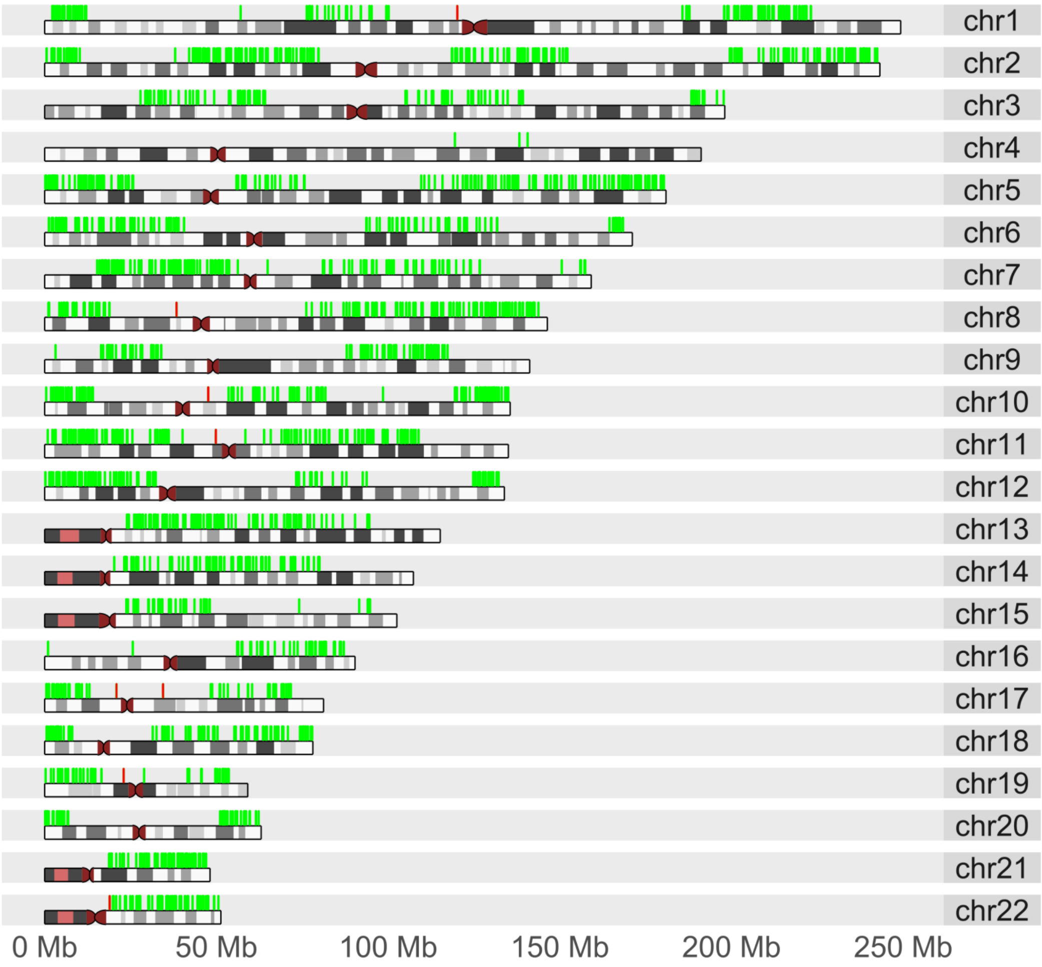


Figure 6.3. Results of gene conversion detection on a trio from in-house oocyte collection. Green – gene conversions detected by GenCall (QC015); red – detected gene conversion after data filtering with SureTypeSC.

6.4 Detection of copy number variants

To this end, we assumed that the cells have correct numbers of chromosomes.. However, aneuploidies and copy number variants (CNVs) add an extra level of uncertainty to the data. Aneuploidy is presence of an abnormal number of chromosomes in a cell. In context of female meiosis, it is estimated that 5 % of pregnancies are affected by this chromosome abnormality, which can be fatal for the fetus or cause major genetic disorders (Hassold and Hunt, 2001). CNVs are conditions where only part of the chromosome is deleted or duplicated. Similarly to aneuploidies, they can have fatal consequences to the female oocyte (Martin et al., 2015).

We were interested whether SureTypeSC would improve biological insight when used for high precision (RF-GDA with stringent threshold. We assessed copy number variants (CNVs) in human oocyte data (Ottolini et al., 2015) and successfully resolved chromosomal deletion. The loss of a chromosome or chromosome segment results in one cell with only A and B calls (no heterozygous SNPs). The loss, however, is obscured by ADI when using the standard GenCall algorithm (Figure 6.4). SureTypeSC removes the ADIs (erroneous AB), increasing the certainty of the inference (Vogel et al., 2019).

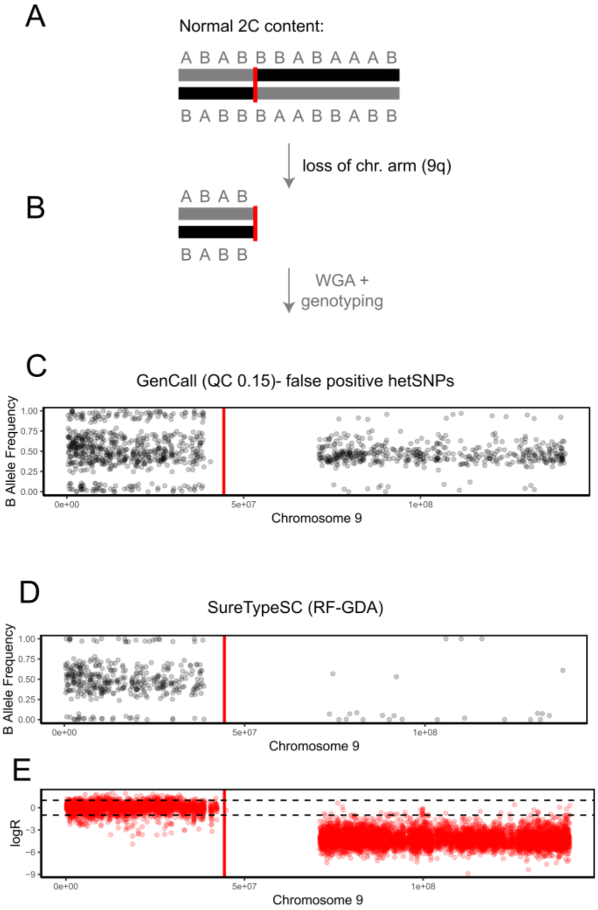


Figure 6.4. Segmental aneuploidy in a human oocytes. (A) The oocyte (Model I) is diploid, but contains a partial deletion (the entire q arm). (B) After WGA, the genotyping algorithm should reject the signal noise from SNPs on the q arm. However, the B allele frequency of genotyped SNPs passed the quality control of GenCall (C). In contrast, the B allele frequency plot of SureTypeSC (D) correctly rejects genotypes from the deleted chromosome arms calling only a few SNPs.

(E) shows the logarithmic ratio of observed vs. expected signal intensities along the chromosome. It confirms that the area of the chromosomal loss has lower signal (negative logR).

6.5 Detecting subpopulations in the single cells

6.5.1 Introduction

To this end, we assumed that all variants in the cell line GM12878 (Chapter 3) that do not match the reference genome (`erroneous variants`) are allele drop outs or allele drop ins (Figure 6.5A). However, it has been previously reported, that the populations of cells of the same type are often heterogeneious (Altschuler and Wu, 2010). We were curious whether some of erroneous variants could be real and therefore used to detect heterogeneity within a cell population. We were interested in whether we could use SureTypeSC with high precision to reveal biological variability within the tested cell line GM12878.

6.5.2 Results

We performed high precise genotype filtering using GenCall and RF-GDA (SureTypeSC) of 46 cells from GM12878 (Section 3.2). Both algorithms were adjusted for high recall (GenCall score 0.87, SureTypeSC score 0.75). To explore the heterogeneity in terms of concordance with the reference bulk DNA genome, we coded variant genotypes that corresponded the reference with one and variants not matching the reference with zero. Similarly to Zafar et al., 2016, we imputed the missing genotypes with value 0.5 and used Hamming distance to calculate the pairwise dissimilarity between the 46 cells in GM12878. We then performed hierarchical clustering on all SNPs that contained at least one non-reference variant across the 46 cells. We included raw genotypes (QC001) ,genotypes from stringent GenCall (0.87), and SureTypeSC (0.75) and used Ward distance for hierarchical clustering. We assessed stability of the clusters by performing bootstrap analysis implemented in the fpc package in R. We subsequently labelled the clusters with clusterwise Jaccard bootstrap mean (calculated from 1000 replicates) indicating the stability of the cluster (Figure 6.5).

The hierarchical clustering reveals there are potentially four subpopulations of cells in GM12878 cell line that are invariant to the type of filtration used (Figure 6.5B, C, D). The bootstrap analysis, however, reveals that only the RF-GDA consistently gives four stable subpopulations (Jaccard mean bootstrap value for a cluster > 0.75, Hennig 2007).

The unstable clusters present in the trees from the minimal filter (QC 0.01) and ‘high precision’ genotyping using GenCall suggest non-reproducible noise being transferred to the bootstrapped replicates that is removed by SureTypeSC. Using a high precision mode, SureTypeSC, but not GenCall, was able to stably detect four subpopulations in the reference GM12878 cell line. Thus, SureTypeSC most likely revealed true heterogeneity within the single cell population (Vogel et al., 2019).

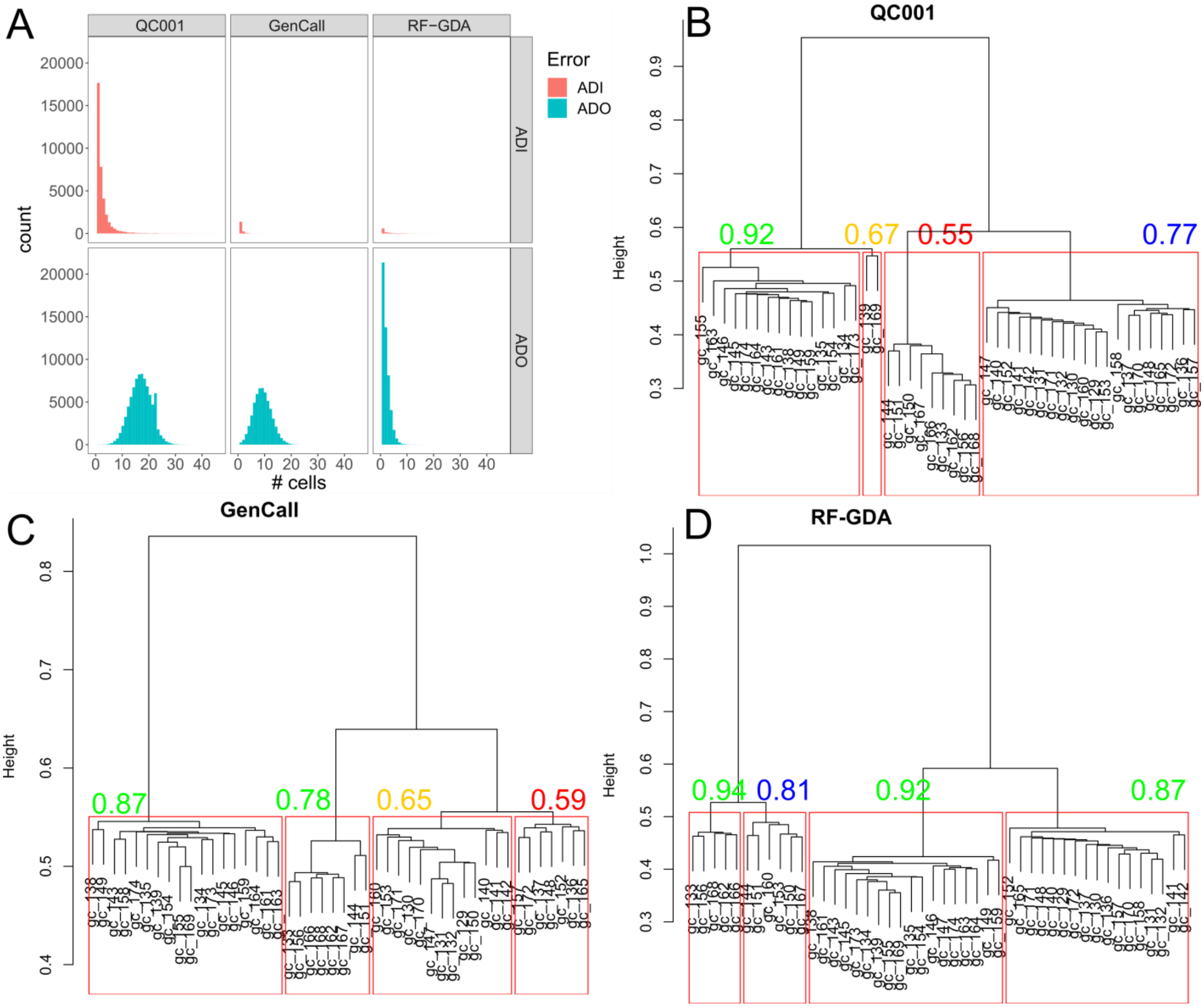


Figure 6.5. Detected errors and potential variants in the 46 cells of GM12878. (A) Histogram of detected ADI and ADO in raw data (left panel), data filtered with GenCall at high precision (GenCall score 0.87, middle panel) and data filtered with SureTypeSC (right panel). Hierarchical clustering on raw data (B), on data from GenCall (C) and from RF-GDA (SureTypeSC, D). The histograms were generated in R using hclust and evaluated using the clusterboot function from the R package *fpc*. The red rectangles show the potential subpopulations and the numbers indicate the Jaccard bootstrap mean. Labels are coloured according to the following rules (obtained from Henning 2007 and manual of the *fpc* package): Green labels indicate highly stable clusters (>0.85), blue labels indicate stable clusters (>0.75); orange labels might indicate a pattern, however the membership of the cells to a particular cluster is doubtful; red labels (<0.6) indicate unstable clusters. RF-GDA is the only algorithm that gives four stable clusters.

7 Conclusion

The rapid evolution of single cell genomics is constantly reinforced by hundreds of exciting discoveries about how our body works on a fine level of single cell. A lot of knowledge from various fields of biology is required to explain how cancer cells develop, how cells in our brain communicate or which factors contribute to successful pregnancy. Precise genetic information about a cell is one of the pieces to the puzzle.

Accurate detection of genotype of a single cell is still challenging due to preciousness of the genetic material. While the laboratory techniques are trying to improve the methodology to deliver more accurate signal, it is a matter of fact that the whole genome amplification is the bottleneck of the whole process.

This work targets an important issue of single cell bioinformatics – how to reliably distinguish between a genotype artefact and real signal in the single-cell data. While this problem has been solved algorithmically for NGS data by various approaches, the SNP array technology has been left behind. This is unfortunate, because SNP array offers a cost-efficient alternative for analysing thousands of genomic loci with good coverage, which is confirmed by accuracy and call rate over 99% for standard, bulk DNA.

In this work, a reference population of single cell samples from SNP array was gathered and the noise that is coming through when using the state of the art genotyping workflow for SNP arrays was analysed. A cascade machine learning method that learns the pattern of noise in the single cell data was developed. Thorough validation of the method reveals that it is possible to recall more single-cell genotypes with better precision than with the current state of the art represented by the GenCall algorithm.

Furthermore, algorithms for knowledge extraction from products of female meiosis were designed and optimized, particularly for crossover detection. The improved genotype detection has direct impact on quality of the knowledge gathered from the data. It is likely that the presented algorithms can improve both, diagnostics and biological inference.

SureTypeSC was tested on a reference population of single cells. As these cells were clones, one would expect that the only source of heterogeneity is random noise. It however turns out, that after noise removal, the cells still contain differences and create subpopulations. This confirms what mentioned at the very beginning of this work – heterogeneity is everywhere within our body. Ability of deciphering of subpopulations in the single cell data would have a likely application in cancer biology.

7.1 Summary of contribution

As mentioned at the beginning, this work was motivated by research interests of two scientific groups that I was member of during my doctoral studies. These groups supported my bioinformatics research with databases of biological data. The biological conclusions were inferred based on evidence found by methods presented in this work. Related to research questions asked in Section 1.2.1, I summarize the contribution of this thesis in the following points:

- I generated a reference population of single cells from SNP arrays with a reliable genotype confirmed by information from bulk DNA. This is a valuable resource for future research serving as a training dataset
- I identified pattern of noise in the data and based on this, I designed, implemented and validated an original two stage machine learning method named SureTypeSC. The method evaluates a single-cell genotype from SNP array and calculates a confidence measure. The algorithm outperforms current state of the art in genotyping of single-cell SNP array data
- I designed and implemented algorithms for crossover detection in single-cell data from female meiosis. I demonstrated application and benefits of using SureTypeSC with these algorithms. I furthermore showed how SureTypeSC can help to solve other relevant questions of single-cell bioinformatics: aneuploidy detection, subpopulations clustering and gene conversion analysis.

Bibliography

Altschuler, S.J., and Wu, L.F. (2010). Cellular Heterogeneity: Do Differences Make a Difference? Cell *141*, 559–563.

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.

Bovik, A.C., Huang, T.S., and Munson, D.C. (1987). The Effect of Median Filtering on Edge Estimation and Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence *PAMI-9*, 181–194.

Breiman, L. (2001). Random Forests. Mach. Learn. *45*, 5–32.

Chen, J.-M., Cooper, D.N., Chuzhanova, N., Férec, C., and Patrinos, G.P. (2007). Gene conversion: mechanisms, evolution and human disease. Nature Reviews Genetics *8*, 762–775.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. Ser. B *39*, 1–38.

Genton, M.G. (2002). Classes of Kernels for Machine Learning: A Statistics Perspective. J. Mach. Learn. Res. *2*, 299–312.

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.

Han, J., Kamber, M., and Pei, J. (2011). Data Mining: Concepts and Techniques, Third Edition (Haryana, India; Burlington, MA: Morgan Kaufmann).

Hassold, T., and Hunt, P. (2001). To err (meiotically) is human: the genesis of human aneuploidy. Nature Reviews Genetics *2*, 280.

Hastie, T., Tibshirani, R., and Friedman, J. (2016). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (New York, NY: Springer).

Krishnamoorthy, K. (2006). Handbook of Statistical Distributions with Applications (Boca Raton: Chapman and Hall/CRC).

Martin, C.L., Kirkpatrick, B.E., and Ledbetter, D.H. (2015). CNVs, Aneuploidies and Human Disease. Clin Perinatol *42*, 227–242.

Ng, A. (2019). CS229 Lecture Notes, PartIV: Generative Learning algorithms.

Ottolini, C.S., Newnham, L., Capalbo, A., Natesan, S.A., Joshi, H.A., Cimadomo, D., Griffin, D.K., Sage, K., Summers, M.C., Thornhill, A.R., et al. (2015). Genome-wide maps of recombination and chromosome segregation in human oocytes and embryos show selection for maternal recombination rates. Nat. Genet. 47, 727–735.

Ottolini, C.S., Capalbo, A., Newnham, L., Cimadomo, D., Natesan, S.A., Hoffmann, E.R., Ubaldi, F.M., Rienzi, L., and Handyside, A.H. (2016). Generation of meiomaps of genome-wide recombination and chromosome segregation in human oocytes. Nat Protoc *11*, 1229–1243.

Padhukasahasram, B., and Rannala, B. (2013). Meiotic gene-conversion rate and tract length variation in the human genome. European Journal of Human Genetics.

Ritchie, M.E., Liu, R., Carvalho, B.S., Irizarry, R.A., and The Australia and New Zealand Multiple Sclerosis Genetics Consortium (ANZgene) (2011). Comparing genotyping algorithms for Illumina's Infinium whole-genome SNP BeadChips. BMC Bioinformatics *12*, 68.

Shapiro, E., Biezuner, T., and Linnarsson, S. (2013). Single-cell sequencing-based technologies will revolutionize whole-organism science. Nature Reviews Genetics *14*, 618–630.

Staaf, J., Vallon-Christersson, J., Lindgren, D., Juliusson, G., Rosenquist, R., Höglund, M., Borg, Å., and Ringnér, M. (2008). Normalization of Illumina Infinium whole-genome SNP data improves copy number estimates and allelic intensity ratios. BMC Bioinformatics *9*, 409.

Vogel, I., Blanshard, R.C., and Hoffmann, E.R. SureTypeSC—a Random Forest and Gaussian mixture predictor of high confidence genotypes in single-cell data. Bioinformatics.

Vogel, I. Single-cell genotyping. Brno, 2019. PhD thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Ing. Jaroslav Zendulka, CSc.

Walt, S. van der, Colbert, S.C., and Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. Computing in Science Engineering *13*, 22–30.

Williams, A.L., Genovese, G., Dyer, T., Altemose, N., Truax, K., Jun, G., Patterson, N., Myers, S.R., Curran, J.E., Duggirala, R., et al. (2015). Non-crossover gene conversions show strong GC bias and unexpected clustering in humans. ELife *4*, e04637.

Zafar, H., Wang, Y., Nakhleh, L., Navin, N., and Chen, K. (2016). Monovar: single-nucleotide variant detection in single cells. Nat. 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.

Curriculum Vitae

Personal information

Full name: Ivan Vogel
Nationality: slovak
Date of birth: 11.7.1986
Email: ivan.vogel@gmail.com

Education

2011 - now	**Faculty of Information Technology**, Brno University of Technology Doctoral degree programme Computer Science and Engineering
2008 – 2011	**Faculty of Information Technology**, Brno University of Technology. Masters degree programme Bioinformatics and Biocomputing
2005 – 2008	**Faculty of Information Technology**, Brno University of Technology. Bachelor’s degree programme Information Technology

Internships and academic stays:

02/2014 – 04/2014	**University of Helsinki**, Insitute of Biotechnology –Schulman Lab Topic: Computational annotation of transposable elements in plants
02/2010 – 06/2010	**Graz University of Technology**, Faculty of Computer Science and Biomedical Engineering. Topic: Study courses in area of software engineering

Honours and awards

10/2014	**Best poster award**. Methods in Plant Sciences. National Conference in Sec, Czech Republic
6/2011	Dean’s award for outstanding Master Thesis.

Employment summary

10/2015 – now	**Center Bioinformatician** – Center for Chromosome Stability, Department of Molecular Medicine, University of Copenhagen, Denmark

01/2012 – 01/2015	**Bioinformatician** - Department of Plant developmental genetics, Institute of Biophysics of the Czech Academy of Sciences, Brno, Czech Republic
01/2013 – 12/2014	**Bioinformatician**, Center European Institute of Technology, Brno, Czech Republic
02/2008 – 12/2008	**Programmer – Analyst**, Asseco Central Europe, a.s., Brno, Czech Republic

Teaching activities

2011 – 2015	**Bioinformatics practices :** lecturing practical exercices and supervising term project
2011 – 2012	**Web Design:** lecturing practical exercises and supervising projects
2011 – 2015	**Supervision** of 1 bachelor thesis and 3 diploma theses

Talks and presentations at conferences and workshops

09/2016	**The Students and Postdocs Meiosis Workshop**, Montpellier, France (oral presentation)
10/2014	**Methods in Plant Sciences**, National Conference in Sec (poster)
09/2013	**Core Facility Genomics Workshop**: Next-gen sequencing technologies and applications (oral presentation)
04/2012	**Student conference EEICT**, Brno, Czech Republic

International courses

03/2012	**Winter school in methods in bioinformatics**, Universitat Rovira i Virgili, Tarragona, Spain

Publications in international journals

Vogel, I., Blanshard, R.C., and Hoffmann, E.R. SureTypeSC—a Random Forest and Gaussian mixture predictor of high confidence genotypes in single-cell data. Bioinformatics.

Kubat, Z., Zluvova, J., Vogel, I., Kovacova, V., Cermak, T., Cegan, R., Hobza, R., Vyskot, B., and Kejnovsky, E. (2014). Possible mechanisms responsible for absence of a retrotransposon family on a plant Y chromosome. New Phytol. *202*, 662–678.

Steflova, P., Tokan, V., Vogel, I., Lexa, M., Macas, J., Novak, P., Hobza, R., Vyskot, B., and Kejnovsky, E. (2013). Contrasting Patterns of Transposable Element and Satellite Distribution on Sex Chromosomes (XY1Y2) in the Dioecious Plant Rumex acetosa. Genome Biol Evol *5*, 769–782.

Larsen, N.B., Liberti, S.E., Vogel, I., Jørgensen, S.W., Hickson, I.D., and Mankouri, H.W. (2017). Stalled replication forks generate a distinct mutational signature in yeast. Proc. Natl. Acad. Sci. U.S.A. *114*, 9665–9670.

Kralova, T., Cegan, R., Kubat, Z., Vrana, J., Vyskot, B., Vogel, I., Kejnovsky, E., and Hobza, R. (2014). Identification of a novel retrotransposon with sex chromosome-specific distribution in Silene latifolia. Cytogenet. Genome Res. *143*, 87–95.

Ren, L., Chen, L., Wu, W., Garribba, L., Tian, H., Liu, Z., Vogel, I., Li, C., Hickson, I.D., and Liu, Y. (2017). Potential biomarkers of DNA replication stress in cancer. Oncotarget *8*, 36996–37008.

Soukupova, M., Nevrtalova, E., Cížková, J., Vogel, I., Cegan, R., Hobza, R., and Vyskot, B. (2014). The X chromosome is necessary for somatic development in the dioecious Silene latifolia: cytogenetic and molecular evidence and sequencing of a haploid genome. Cytogenet. Genome Res. *143*, 96–103.

Abstract

Single-cell genotyping is a challenging part of genomics that deals with insufficient amount of DNA for analysis and methods that amplify the single cell DNA introduce bias in the data. This thesis maps the state of the art of bioinformatics analysis in genomics, particularly SNP array genotyping and proposes and implements original method for tackling the noise in the single-cell data. Moreover, few original algorithms for knowledge extraction from single-cell data are presented and the functionality of the workflow is demonstrated on real data from products of female meiosis.