Računalno određivanje podtipova mišjih tumora dojke

Ivanković, Ivna

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:819133

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

zir.nsk.hr

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

UNIVERSITY OF ZAGREB
FACULTY OF SCIENCE
DIVISION OF BIOLOGY

Ivna Ivanković

Computational Subtyping of Mouse Breast Tumors

Graduation Thesis

Zagreb, 2019

This graduation thesis was created at German Cancer Research Center, Division of Molecular Biology, under the supervision of dr. sc. Marc Zapatka, and submitted for evaluation to the Division of Biology, Faculty of Science, University of Zagreb for the purpose of obtaining a Master of Science Degree in Molecular Biology.

To my supervisor Marc Zapatka for the opportunity, great ideas and a wonderful leadership. To Rosa Karlić for all the constructive advice and generous help. To my beautiful family, supportive friends, and my dearest Ivan.

TEMELJNA DOKUMENTACIJSKA KARTICA
Sveučilište u Zagrebu
Prirodoslovno-matematički fakultet
Biološki odsjek Diplomski rad

RAČUNALNO ODREĐIVANJE PODTIPOVA MIŠJIH TUMORA DOJKE

Ivna Ivanković
Rooseveltov trg 6, 10000 Zagreb, Hrvatska

Istraživanja genske ekspresije tumora dojke metodom mikročipova ukazala su na različite molekularne portrete tumora na temelju kojih ih je moguće klasificirati u pet podtipova s različitim kliničkim ishodima: luminalni A, luminalni B, obogaćeni receptorom HER2, bazalni i tip nalik normalnom. PAM50 je molekularni klasifikator nastao reduciranjem proširenog seta gena na 50 gena koji najznačajnije pridonose prepoznavanju podtipova. U paketu *genefu* programskog jezika R implementirani su bioinformatički algoritmi i genski potpisi za određivanje molekularnih podtipova tumora dojke, uključujući i molekularni PAM50 klasifikator. U ovom diplomskom radu modificiran je algoritam iz paketa *genefu* kako bi ulazni podaci bili podaci dobiveni sekvenciranjem RNA umjesto podaci o genskoj ekspresiji dobiveni metodom mikročipova i kako bi odredio podtipove mišjih tumora u odnosu na ljudske tumore dojke. Nadalje, nedavna istraživanja pokazala su da je moguće integrirati mišje i ljudske setove podataka i tako informacije iz jednog seta koristiti za interpretaciju drugog. U ovom radu primijenjena je kanonska korelacijska analiza iz paketa *Seurat* za integraciju mišjih i ljudskih podataka dobivenih sekvenciranjem RNA tumora dojke i određeni su podtipovi na temelju genskog PAM50 seta. Uspoređeni su rezultati određivanja podtipova mišjih tumora dojke dobiveni paketima *genefu* i *Seurat*, i procijenjena je točnost dvaju neovisnih pristupa određivanja podtipova tumora dojke.

(36 stranica, 7 slika, 4 tablice, 34 literaturna navoda, jezik izvornika: engleski)

Rad je pohranjen u Središnjoj biološkoj knjižnici
Ključne riječi: karcinom dojke, heterogeničnost tumora, PAM50, klasteriranje
Voditelj: dr. sc. Marc Zapatka
Suvoditelj: doc. dr. sc. Rosa Karlić
Ocjenitelji: doc. dr. sc. Sofia Ana Blažević
doc. dr. sc. Romana Gračan
doc. dr. sc. Rosa Karlić
Zamjena: prof. dr. sc. Biljana Balen
Rad prihvaćen: 19. lipnja 2019.

BASIC DOCUMENTATION CARD
University of Zagreb
Faculty of Science
Division of Biology Graduation Thesis

COMPUTATIONAL SUBTYPING OF MOUSE BREAST TUMORS

Ivna Ivanković
Rooseveltov trg 6, 10000 Zagreb, Croatia

Studies of gene expression patterns of breast tumors derived from cDNA microarrays reported their distinctive molecular portraits according to which tumors can be classified into five intrinsic subtypes with distinct clinical outcomes: luminal A, luminal B, HER2 enriched, basal and normal-like. PAM50 is a molecular classifier developed by minimizing expanded intrinsic gene set to the top 50 genes that contribute to distinguishing intrinsic subtypes. The R/Bioconductor package genefu implements bioinformatics algorithms and gene signatures for molecular subtyping of breast cancer, including PAM50 molecular classifier. In this project, an algorithm from package genefu was modified to subtype breast tumors using RNA-Seq instead of microarray data as an input and then used to subtype RNA sequenced mouse breast tumors in relation to human tumors. Furthermore, a recent study showed that human and mouse data can be integrated using canonical correlation analysis from package Seurat. The motivation for integrating diverse datasets lies in potential to use information from one dataset for the interpretation of another. In this project canonical correlation analysis was also used to integrate human and mouse bulk RNA-Seq data based on the set of PAM50 genes and used to determine intrinsic breast tumor subtypes. Finally, results of genefu and Seurat subtyping were compared and the performance of the two independent approaches was assessed.

(36 pages, 7 figures, 4 tables, 34 references, original in: English)

Thesis deposited in the Central Biological Library
Key words: breast cancer, tumor heterogeneity, PAM50, intrinsic subtypes, clustering
Supervisor: dr. sc. Marc Zapatka
Cosupervisor: doc. dr. sc. Rosa Karlić
Reviewers: doc. dr. sc. Sofia Ana Blažević
doc. dr. sc. Romana Gračan
doc. dr. sc. Rosa Karlić
Substitution: prof. dr. sc. Biljana Balen
Thesis accepted: 19 June 2019

CONTENTS

1. Introduction **1**

1.1. Heterogeneity of breast cancer 1

1.2. Measuring gene expression 2

1.2.1. Microarrays 2

1.2.2. RNA Sequencing 3

1.3. R packages 5

2. Aims and Objectives **6**

3. Materials and Methods **7**

3.1. Data retrieval 7

3.2. Genefu subtyping 9

3.2.1. Transformation 9

3.2.2. Batch effect correction 9

3.2.3. Centroid Calculation 10

3.2.4. Subtyping 10

3.2.5. Cross-validation 10

3.3. Seurat subtyping 11

3.3.1. Integration of human and mouse dataset 11

3.3.2. Subtyping 12

3.4. Assessment and comparison 13

3.4.1. Hierarchical clustering 13

3.4.2. Principal component analysis 14

3.4.3. Comparison of genefu and Seurat based subtyping 14

4. Results **16**

4.1. Genefu subtyping 16

4.1.1. Batch correction 16

4.2. Seurat subtyping 17

4.3. Assessment and comparison 18

4.3.1. TCGA data 18

4.3.2. Genefu and Seurat subtyping of human tumors 19

4.3.3. Genefu and Seurat subtyping of mouse tumors 20

5. Discussion **28**

6. Conclusion **31**

References **32**

ABBREVIATIONS

PAM50	Prediction analysis of microarray 50
TCGA	The Cancer Genome Atlas
GDC	Genomic Data Commons
DKFZ	Deutsches Krebsforschungszentrum, German Cancer Research Center
NGS	Next generation sequencing
cDNA	Complementary deoxyribonucleic acid
cRNA	Complementary ribonucleic acid
mRNA	Messenger ribonucleic acid
pre-mRNA	Precursor mRNA
ncRNA	Non-coding RNA
SSP	Single sample predictor
SNN	Shared nearest neighbors
BRCA	Breast cancer
PCC	Pearson correlation coefficient
ER	Estrogen receptor
PR	Progesterone receptor
HER2	Human epidermal growth factor receptor 2

1. Introduction

1.1. Heterogeneity of breast cancer

Breast cancer is the most frequently diagnosed malignancy and the leading cause of carcinoma deaths in women (Siegel et al., 2016). It is not a single disease but rather a diverse group of heterogeneous lesions characterized by distinct pathological types with different clinical outcomes. One of the initial molecular profiling studies of primary tumors conducted by Perou et al. (2000) showed that breast cancer could be segregated into several biologically distinct subtypes. Later, a study of gene expression patterns of breast cancers derived from cDNA microarrays conducted by Sørlie et al. (2001) reported their distinctive molecular portraits according to which tumors were classified into five intrinsic subtypes with distinct clinical outcomes: luminal A, luminal B, HER2-enriched, basal-like, and normal-like. Gene expression profiling studies have given us insight into the heterogeneity of breast tumors and can be used to provide prognostic information beyond standard clinical assessment. A great portion of breast tumors is diagnosed at an early stage and treated with aim to eliminate all tumor cells, but in approximately 30% of women (Colleoni et al., 2016) cancer recurs after initial treatment. To reduce the rate of relapse, adjuvant treatment guided by single biomarkers such as estrogen and progesterone receptors and HER2 was introduced (Harris et al., 2016). Nonetheless, a large fraction of women who would not have relapsed were unnecessarily treated with consequent morbidities and increased cost for the health system (Cardoso et al., 2016). To avoid unnecessary or ineffective treatment and improve patient outcome, guided clinical decisions that take into account the tumor

genomic profile are introduced (Schmidt, 2016).

1.2. Measuring gene expression

Differences underlying the gene expression patterns among breast cancer subtypes reflect the fundamental differences of the tumors at the molecular level (Sørlie et al., 2003). Therefore, multiple expression-based classifiers have been developed and are clinically used to stratify patients with breast cancer, and add significant prognostic and predictive information to standard parameters. One of them is a 50-gene subtype predictor (Parker et al., 2009) that incorporates the gene expression-based intrinsic subtypes luminal A, luminal B, HER2-enriched, basal-like, and normal-like. It was developed by minimizing the intrinsic gene set defined in previous microarray studies to the top 50 genes, PAM50 gene expression signature, that contribute to distinguishing intrinsic subtypes. In addition to mouse models (Park et al., 2018) and microarrays, development of high throughput sequencing contributed to our understanding of breast cancer and RNA sequencing became a widely used method to study gene expression patterns.

1.2.1. Microarrays

Microarrays have revolutionized breast cancer research by enabling studies of gene expression on a transcriptome-wide scale (Fumagalli et al., 2014). With this method thousands of transcripts can be quantified simultaneously which is useful for determining differences in transcript levels under different experimental conditions or disease states. Microarrays consist of thousands of DNA oligonucleotides with known sequence, called probes, printed in a high density array on a glass slide. The core principle behind microarrays is hybridization between two complementary DNA strands. Two RNA samples of interest are reversely transcribed into cDNA and labelled using red and green fluorescent dyes, and then hybridized with the arrayed probes if they contain complementary sequence. After hybridization under specific conditions, non-

complementary cDNA is washed off and the chip is scanned to measure the relative abundance of spotted DNA sequences. Data for each gene consists of two fluorescence intensity measurements (R, G) showing the expression level of the gene in the red and green labelled mRNA samples. The ratio of the fluorescence intensity for each spot represents the relative abundance of the corresponding transcript. In the microarray experiment, however, many undesirable systematic variations that affect the measured gene expression levels are observed. To remove those sources of variation, normalization methods are applied. The main idea of normalization is to adjust for artifactual differences in intensity of the two labels such as: affinity of the two labels for DNA, amounts of sample and label used, differences in photomultiplier tube and laser voltage settings, differences in photon emission response to laser excitation (Park et al., 2003).

1.2.2. RNA Sequencing

High-throughput next-generation sequencing (NGS) enabled RNA analysis which provides, in contrary to low-throughput methods such as Northern blots and quantitative polymerase chain reaction, a more detailed and quantitative view of gene expression. In addition, this method emerges as a superior alternative to microarrays to define gene expression levels (Wang et al., 2009). High levels of background noise arising from non-specific hybridization and probe saturation affect the quantification of transcripts expressed at low and high levels, limit the dynamic range of the microarray technology (Fumagalli et al., 2014) whereas RNA-Seq technology efficiently addresses this issue. In addition, novel RNA transcripts can be detected because, unlike microarrays, RNA-Seq technology does not require transcript-specific probes.

A typical RNA-Seq experiment consists of isolating RNA, preparing the sequencing library, and sequencing it on an NGS platform. After the extraction of RNA from a biological sample, RNA-Seq library is created by isolating the desired RNA molecules, reverse-transcribing the RNA to complementary DNA (cDNA), fragmenting or amplifying randomly primed cDNA molecules, and ligating sequencing adaptors. There

are several different library designs in RNA-Seq library protocols. Desired RNAs, in this case mRNA, can be isolated from the mixture of all extracted RNAs from the cell using various methods, including poly-A selection which selects for RNA species with poly-A tail and enriches for mRNA, and ribo-depletion which enriches for mRNA, pre-mRNA and ncRNA by depleting ribosomal RNA. Desired single-stranded RNAs are then converted to double-stranded cDNA using strand-specific or non-strand-specific protocol. Strand-specific protocols allow an assignment of the reads to their original strand by attaching distinct adapters in a known orientation relative to the 5' and 3' ends of the original mRNA. Non-strand-specific protocols are cheaper and less time consuming, but they do not provide an information if a read originated from the sense or antisense strand of the reverse transcribed mRNA. After converting selected RNAs to cDNA, adaptors are ligated to the ends of cDNA fragments, given fragments are then amplified by polymerase chain reaction (PCR), and produced RNA-Seq library is sequenced. Following typical RNA-Seq experiment, reads are aligned to a reference genome and the expression level of each gene is estimated by counting the number of reads that align to each transcript or exon (Kukurba and Montgomery, 2015). To accurately estimate gene expression and detect differential expression, read counts must be normalized to correct for systematic variability such as read depth, library fragment size, and sequence composition bias (Oshlack and Wakefield, 2009). To analyse differential expression, a variety of statistical methods to account for the specifics of count data, such as non-normality and a dependence of the variance on the mean, have been designed and implemented in R packages. DESeq2 package for differential analysis of count data uses shrinkage estimators for variances (or, equivalently, dispersions) and fold change to improve stability and interpretability of estimates which results with a more quantitative analysis focused on the strength rather than the mere presence of differential expression (Love et al., 2014). For RNA-Seq data, the problem of heteroskedasticity arises: on the original count scale variances are strongly dependent on the mean counts, and the result is dominated by highly expressed and highly variable genes. Therefore, it is useful to transform count data before the analysis. Va-

riance stabilizing transformation (VST) implemented in DESeq2 package is effective at stabilizing variance. It yields a matrix of homoskedastic values, whose variances are approximately the same throughout the dynamic range, that are normalized with respect to library size.

1.3. R packages

Packages used and in this research are genefu (Gendoo et al., 2015) and Seurat (Butler et al., 2018). genefu is used for microarray gene expression data analysis in breast cancer studies such as gene mapping between different microarray platforms, identification of molecular subtypes, implementation of published gene signatures, gene selection, and survival analysis. It implements bioinformatics algorithms and gene signatures for molecular subtyping of breast cancer, including single sample predictor (SSP) molecular subtype classification algorithm and PAM50 gene expression signature. SSP is a nearest centroid classifier where the centroids representing the intrinsic subtypes were originally identified through hierarchical clustering using a specific gene list, in this case PAM50 (Sørlie et al., 2003). In subtyping using nearest centroid method each centroid represents one of the five intrinsic subtypes. This method works by measuring the distance of each mouse sample to all centroids and assigning the sample to the closest one.

Seurat is a novel package designed for quality check, and the analysis of single-cell RNA-Seq data. In this package analytical strategy for integrating scRNA-seq data sets based on common sources of variation is introduced, enabling the identification of shared populations across data sets and downstream comparative analysis. Additionally, a computational strategy to integrate diverse datasets together, called Canonical Correlation Analysis, is implemented enabling integration and comparison of single cell measurements not only across scRNA-seq technologies, but different modalities as well (Stuart et al., 2019).

2. Aims and Objectives

Essential approach to examine underlying mechanisms and genetic pathways in breast cancer, as well as create approaches for modeling clinical tumor subtypes, is the use of mouse models. In addition to mouse models, development of high throughput sequencing contributed to understand the breast cancer and RNA sequencing became widely used method to study gene expression patterns.

The aim of this research is to modify PAM50 molecular subtype classification algorithm from package genefu and apply Canonical Correlation Analysis implemented in package Seurat to subtype RNA sequenced mouse breast tumors in relation to human tumors, to compare the results of subtyping mouse breast tumors, and to assess the performance of the two independent approaches for subtyping breast tumors. To modify PAM50 molecular subtype classification algorithm from genefu package, PAM50 gene expression signature will be adapted by calculating new PAM50 centroids. Furthermore, a recent study showed that human and mouse single-cell sequenced data can be integrated using Canonical Correlation Analysis implemented in R package Seurat. The motivation for integrating diverse datasets lies in the potential to use information from one dataset for the interpretation of another. Therefore, CCA from Seurat package will be used to integrate human and mouse bulk RNA sequenced data based on the set of PAM50 genes and to determine intrinsic breast tumor subtypes.

3. Materials and Methods

All the analysis was performed using R version 3.4.4 in the integrated development environment RStudio, code used to produce this research is available in Appendix A, B and C.

3.1. Data retrieval

In this work, gene expression data obtained by RNA sequencing method were analysed. The Cancer Genome Atlas (TCGA) database was used to retrieve RNA sequencing data from human primary breast tumor samples and matched samples from healthy breast tissue. This database contains over 20,000 molecularly characterized primary cancers and matched normal samples spanning 33 cancer types. The human dataset consisting of RNA sequenced primary breast tumors and healthy breast tissue was prepared by downloading raw counts produced from TCGA database using TCGAbiolinks, an R/Bioconductor package for integrative analysis with GDC data (Colaprico et al., 2015). Downloaded TCGA breast tumor samples also contain the clinically determined PAM50 subtype information from the original publication (Network et al., 2012). The human dataset consisted of 16679 ortholog genes and 1186 samples; 1073 samples from primary breast tumor and 113 samples from healthy breast tissue. There are 556 samples clinically subtyped as luminal A, 207 of them as luminal B, 82 HER2-enriched, 188 basal-like, and 40 normal-like. For TCGA Breast Cancer (BRCA) project, mRNAs were isolated using poly-A selection and RNA-Seq library protocol was non-strand-specific.

The mouse dataset used in this work was created by combining mouse breast tumor samples from three different sources: raw counts of RNA sequenced mouse tumors available at Sequence Read Archive (SRA) with identifier SRP115453, RNA sequenced healthy breast tissue from DKFZ, and mouse breast tumor samples from ARCHS4 (Lachmann et al., 2018) database (Table 4.3). Dataset contained the expression values of 16679 ortholog genes for 82 mouse samples; 9 of them are control samples where healthy breast tissue was sequenced, and 73 tumor samples. Samples downloaded from ARCHS4 database initially contained 47 mouse RNA-Seq breast tissue samples from 8 different series. Series that consist of only one or two samples were removed, and all samples from series GSE8138 were also removed because those were mouse xenografts. After removing 4 series, 25 samples from 4 series were appropriate for further analysis. Samples from series GSE85810 are breast tumor organoids (Delaunay et al., 2016), and others are breast tumors. Some of the samples downloaded from ARCHS4 were prepared using non-strand-specific protocol, and some using strand-specific, but all used poly-A method for mRNA selection. Method for mRNA selection in all 9 control samples was ribo-depletion and non-strand-specific protocol was used. Samples downloaded from SRA are non-strand-specific and poly-A method for mRNA selection was used.

Table 3.1: Information about different sources of mouse samples that were combined to produce a single mouse dataset for the analysis and subtyping. Number of samples, types of breast tissue, and methods of library preparation are also provided.

Source	Tissue type	Samples	Selection method	Strand specificity
SRA	Tumor	48	poly-A	no
GSE85810	Tumor organoid	6	poly-A	no
GSE77107	Tumor	6	poly-A	yes
GSE81941	Tumor	4	poly-A	yes
GSE112094	Tumor	9	poly-A	no
DKFZ	Healthy	9	ribo-depletion	no

For further analysis, only mouse orthologs of human genes were selected. To achieve this, mouse gene symbols were converted to human gene symbols using gene mappings that were downloaded from BioMart database using biomaRt package (Durinck et al., 2005). Then, the list of human and mouse orthologs was download from BioMart using the same package.

3.2. Genefu subtyping

In this work PAM50 molecular subtype classification algorithms from genefu were modified for use with RNA-Seq instead of microarray expression data, by calculating new centroids specific to each of the intrinsic subtypes of human TCGA dataset. Then, it was used to subtype mouse tumors into intrinsic subtypes based on newly calculated centroids of PAM50 gene expression signature. To do that, human and mouse datasets were transformed separately, then combined and corrected for organism type. After centroid calculation, cross validation technique was used to assess the performance of the modified algorithm in subtyping of human breast tumor samples.

3.2.1. Transformation

To accurately estimate gene expression, raw counts have to be normalized in order to correct for systematic variability such as library size and read depth. In this work raw counts in human and mouse datasets were separately transformed using variance stabilizing transformation (VST) implemented in R package DESeq2. VST yields a matrix of homoskedastic values, that is values with constant variance along the range of means that are normalized with respect to library size.

3.2.2. Batch effect correction

Prior to VST, checking for outliers in human and mouse datasets was performed by calculating pairwise correlation of quantile normalized gene expression values. Quantile normalization is a non-linear transformation that replaces each feature value with

the mean of the features across all the samples with the same quantile. That way observed distributions of each sample are forced to be the same and the average distribution, obtained by taking the mean of each quantile across samples, is used as the reference (Hicks and Irizarry, 2015). Possible sources of batch effects were identified by researching their protocols for RNA-Seq experiments. After VST, mouse dataset was corrected for library preparation type and strand specificity. Then, mouse and human datasets were merged according to matching ortholog genes, and corrected for organism type to make them comparable.

3.2.3. Centroid Calculation

To calculate PAM50 centroids, human dataset was subsetted from combined and corrected dataset and the information about subtype was added. Only 50 genes that belong to PAM50 gene expression signature were subsetted from the set of human and mouse orthologs. Samples having the same subtype were grouped and new centroids for each subtype were calculated by averaging the expression values for each out of 50 genes.

3.2.4. Subtyping

Newly calculated centroids based on human TCGA breast cancer RNA-Seq data were incorporated into genefu and intrinsic.cluster.predict() function with modified centroids was further used to subtype breast tumor samples. This function identifies the breast cancer molecular subtypes using SSP molecular subtype classification algorithm.

3.2.5. Cross-validation

Cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing the data into two segments: one, train set, used to learn or train a model and the other, test set, used to validate the model (Liu and Özsu, 2009). Overall accuracy of subtyping model with newly calculated centroids was assessed using 10-fold cross-validation on human TCGA dataset. The dataset was randomly split into

10 groups, or folds, of approximately equal size. The first group was hold out presenting the test data set. Centroids were calculated based on the remaining 9 groups, or the train set. Then, subtyping of test dataset was performed using newly calculated centroids from the train set. Model performance was evaluated by calculating the percentage of accurately subtyped samples from the test set considering the information about clinically determined PAM50 subtypes previously downloaded from TCGA.

3.3. Seurat subtyping

In this work, slight changes in parameters of functions CreateSeuratObject() and ScaleData() were made with the aim to use human and mouse bulk RNA-Seq data instead of single cell RNA-Seq data. Then, the two datasets were integrated by Canonical Correlation Analysis (CCA) transferring information from the human reference dataset to subtype mouse tumors.

3.3.1. Integration of human and mouse dataset

Two diverse datasets, human reference dataset and mouse query dataset, were integrated using runCCA() function implemented in Seurat. This function performs Canonical Correlation Analysis (CCA), followed by L2-normalization of the canonical correlation vectors, to project the datasets into a shared space defined by shared correlation structure across datasets (Figure 3.1: A-B). Pairs of mutual nearest neighbours across reference and query datasets are then identified (grey lines, Figure 3.1: C), representing samples in a shared biological state which serve as "anchors" to guide the integration. To solve for the problem of observing incorrect anchors (red lines, Figure 3.1: C), a score is assigned based on the consistency of anchors across the neighborhood structure of each dataset. Incorrect anchors have low scores and are downweighted in further analyses (Figure 3.1: D). At the end, anchors and their scores are utilized to compute "correction" vectors for each query sample, transforming its expression so it can be analysed as part of the reference dataset (Figure 3.1: E).

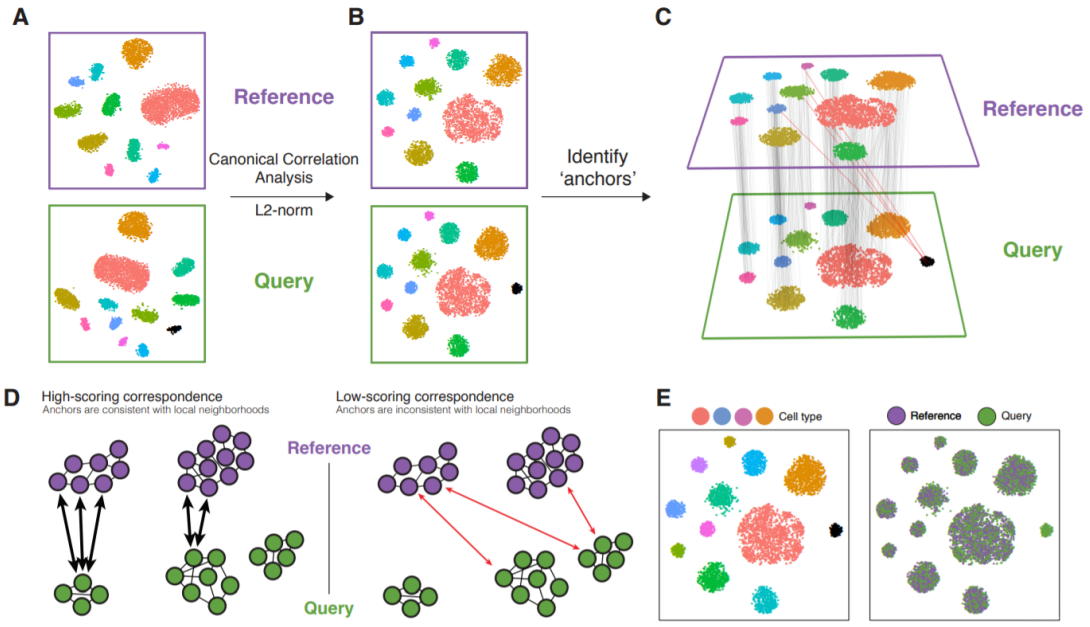

Figure 3.1: Schematic overview of reference "assembly" integration in Seurat (Stuart et al., 2019)

With the calculation of correction vectors, this method does not require previous batch correction as in case with using SSP molecular subtype classification algorithm from genefu. Here, the difference in expression profiles between two samples represents a batch vector and for each cell in the query dataset, a transformation is applied (correction vector) that represents a weighted average across multiple batch vectors (Haghverdi et al., 2018). Weights are determined by a sample similarity score and the anchor score. The sample similarity score is defined by the distance between each query cell and its k nearest anchors in principal component space, prioritizing anchors representing a similar biological state (Stuart et al., 2019).

3.3.2. Subtyping

Two diverse datasets were integrated applying CCA on PAM50 genes of mouse and human datasets consisting of expression values that were previously transformed using VST. The results of integration were visualized using T-distributed Stochastic Neighbor Embedding (t-SNE). This is a technique for embedding high-dimensional data for

visualization in a low-dimensional space; in this work, integration result was visualized in 2D space. The t-SNE algorithm uses local relationships between points to create a low-dimensional mapping. It works by constructing a probability distribution using the Gaussian distribution over pairs of high-dimensional objects with more similar pairs having higher probability of being selected. This distribution defines the relationships between the points in high-dimensional space. Then, the Student t-distribution is used to recreate previously produced probability distribution in low-dimensional space (Maaten and Hinton, 2008). Clusters representing subtypes were calculated using FindClusters() function which identifies clusters of samples by a shared nearest neighbor (SNN) clustering algorithm (Waltman and Van Eck, 2013). In this algorithm, k-nearest neighbors are calculated and the SNN graph is constructed. Then, the modularity function is optimized to determine clusters.

3.4. Assessment and comparison

Before performing assessment of genefu and Seurat based subtyping, TCGA dataset was analysed by applying hierarchical clustering and Principal Component Analysis (PCA) to examine how intrinsic subtypes group. Then, the accuracy of subtyping with genefu was assessed with 10-fold cross-validation. The accuracy of clustering with Seurat was assessed with calculating clusters and visualizing results using t-SNE.

3.4.1. Hierarchical clustering

Clustering is a form of unsupervised learning used to draw inferences from unlabeled data. To cluster TCGA data based on PAM50 genes, agglomerative hierarchical clustering using Pearson's distance measure and average cluster linkage was performed using R function hclust(). Agglomerative hierarchical clustering begins with each point in a distinct cluster, and then combines clusters based on the chosen similarity measure. In average linkage hierarchical clustering, the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other

cluster. The similarity between clusters is usually calculated from the dissimilarity measures. Here Pearson's distance was used, also referred to as the Pearson correlation coefficient (PCC), which is a measure of the linear correlation between two variables. Then, a heatmap with added color label for each intrinsic subtype was drawn to visualize clustering. Heatmap is a color coded table where rows and/or columns are sorted by hierarchical clustering to visually identify patterns. Gene expression data are often visualized that way, where rows represent genes and columns of a heatmap represent samples, and colors represent the intensities of the underlying gene expression.

3.4.2. Principal component analysis

Principal Component Analysis is a dimensionality-reduction method that transforms a large set of correlated variables into a smaller set of uncorrelated variables called principal components. It can be considered as a rotation of the axes of the original variable coordinate system to new orthogonal axes such that the new axes correspond with directions of maximum variation of the original observations (Campbell and Atchley, 1981). Here, PCA was performed on TCGA data choosing PAM50 genes for the calculation, and the data described by the first two principal components were visualized. The first principal component is the direction in space along which projections have the largest variance, and the second is the direction which maximizes variance among all directions orthogonal to the first. That way, dataset is presented in lower-dimensional dorm without losing too much information.

3.4.3. Comparison of genefu and Seurat based subtyping

To assess the accuracy genefu subtyping, 10-fold cross validation was performed as described in section 2.5.5. The accuracy of Seurat subtyping was assessed by subtyping human samples with mouse samples excluded from the analysis. FindClusters() function was used and calculated clusters were compared with clinically determined PAM50 subtypes downloaded from TCGA, and the results were visualized using t-

SNE. Mouse breast tumors were subtyped with genefu and Seurat, and the results were compared.

4. Results

4.1. Genefu subtyping

In this project, PAM50 molecular subtype classification algorithm from genefu package was modified to subtype RNA-Seq data instead of microarray data as an input by manually calculating PAM50 centroids specific to the set of downloaded human breast cancer samples from the TCGA. These centroids were used to subtype RNA sequenced mouse breast tumors in relation to intrinsic subtypes of human breast tumors.

4.1.1. Batch correction

Batch effect in mouse data due to technical differences among samples was detected after calculating pairwise correlation of quantile normalized gene expression values (Figure 4.1, left). Regressing out variables that contain the information about library type and strand specificity, reduced the batch effect between mouse samples from different laboratories (Figure 4.1, right).

The batch effect before correction was greatly present between mouse and human datasets (Figure 4.2, left) with mouse samples showing much lower gene expression in comparison to human samples. After merging human and mouse datasets, and correcting it for organism type, the batch effect was successfully removed (Figure 4.2, right) making the two datasets comparable for biological differences.

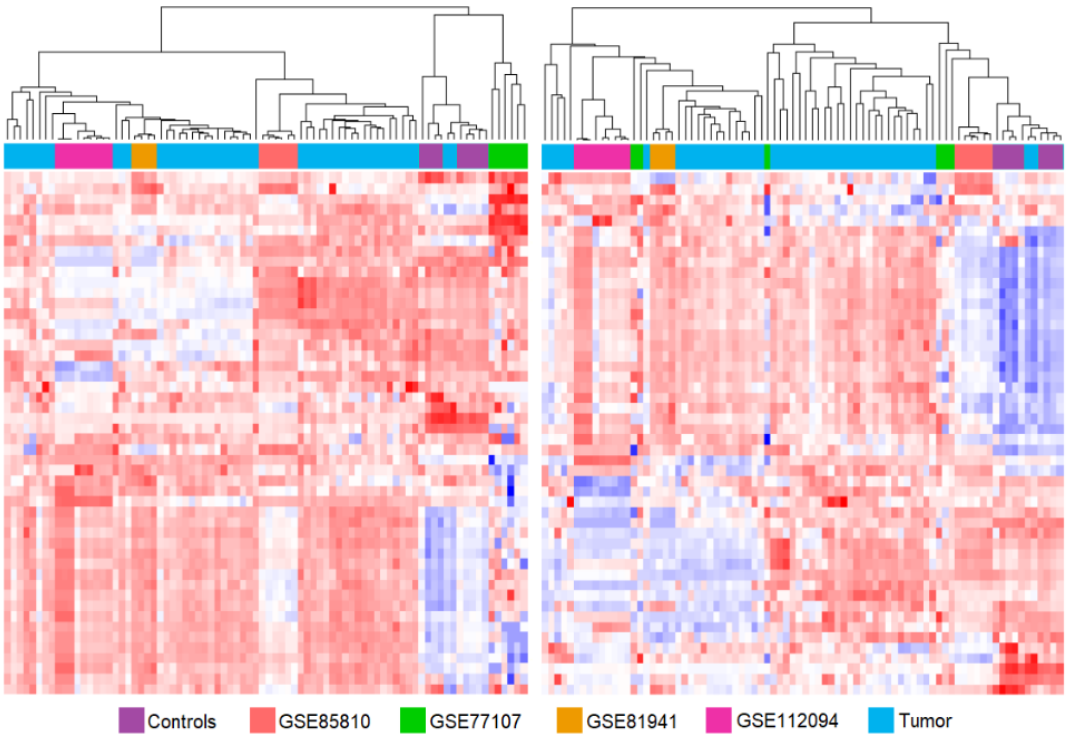

Figure 4.1: Uncorrected (left) and corrected (right) expression values of PAM50 signature in rows for mouse dataset consisting of 82 samples from 6 different series. Mouse dataset was corrected for two variables: library type and strand specificity. Hierarchical clustering was performed using average linkage and Pearson's distance. High levels of expression are in red, low levels of expression are in blue.

4.2. Seurat subtyping

Novel package for the analysis of single-cell RNA-Seq data was applied to analyse bulk RNA-Seq data. CCA was performed to integrate two diverse datasets and mouse samples were integrated across all human samples (Figure 4.3). Information about intrinsic subtypes from human TCGA dataset was used to subtype mouse breast tumor samples (Figure 4.4). Mouse samples are generally not evenly distributed across the human subtypes, but rather group near the edges of each intrinsic subtype indicating successful integration of two datasets with removed technical but preserved biological differences among breast tumors from distinct species.

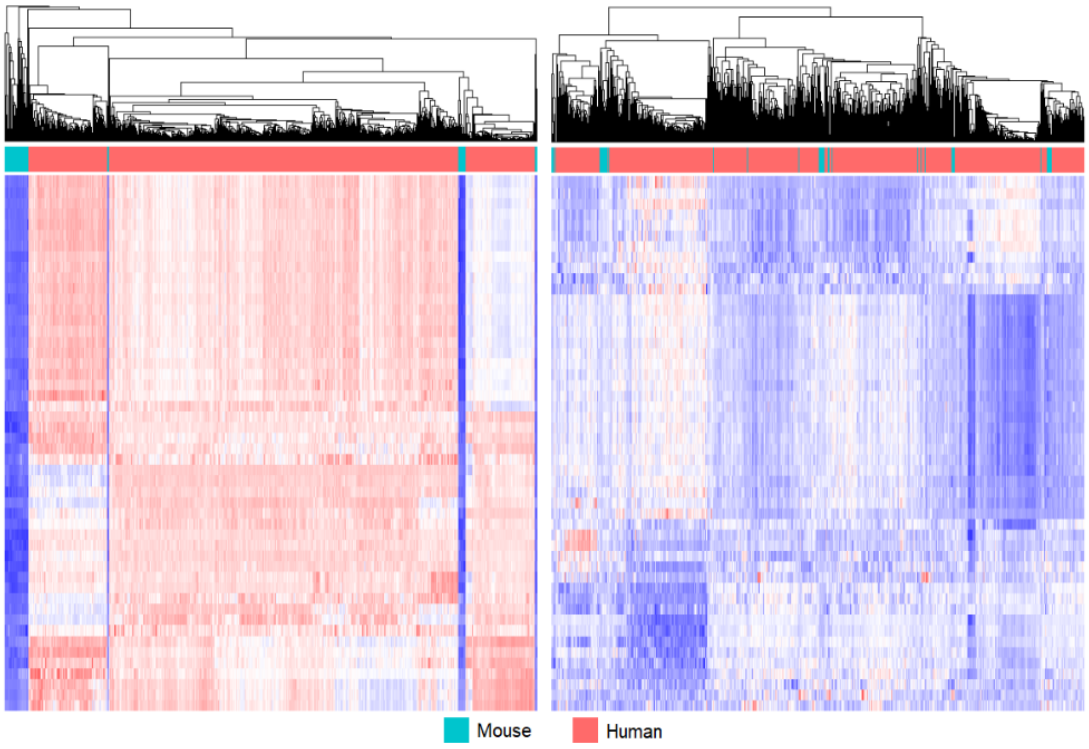

Figure 4.2: Uncorrected (left) and corrected (right) expression values of PAM50 signature in rows for merged dataset consisting of 1186 human TCGA samples and 82 mouse samples. Merged dataset was corrected for organism type. Hierarchical clustering was performed using average linkage and Pearson's distance. High levels of expression are in red, low levels of expression are in blue.

4.3. Assessment and comparison

4.3.1. TCGA data

Prior to adding mouse dataset to human, clustering of TCGA human dataset based on PAM50 genes was performed using hierarchical clustering (Figure 4.5). Hierarchical clustering grouped a great portion of healthy breast tissue samples, basal-like and HER2-enriched breast tumors in separate clusters. Normal-like tumors are clustered mostly among healthy samples, but some are grouped with HER2-enriched and basal-like subtypes. A part of luminal A overlaps with luminal B subtypes (Figure 4.5). Breast cancer samples from TCGA human dataset were also visualized using PCA and colored according to their intrinsic subtype. In PCA plot it can be seen how portion

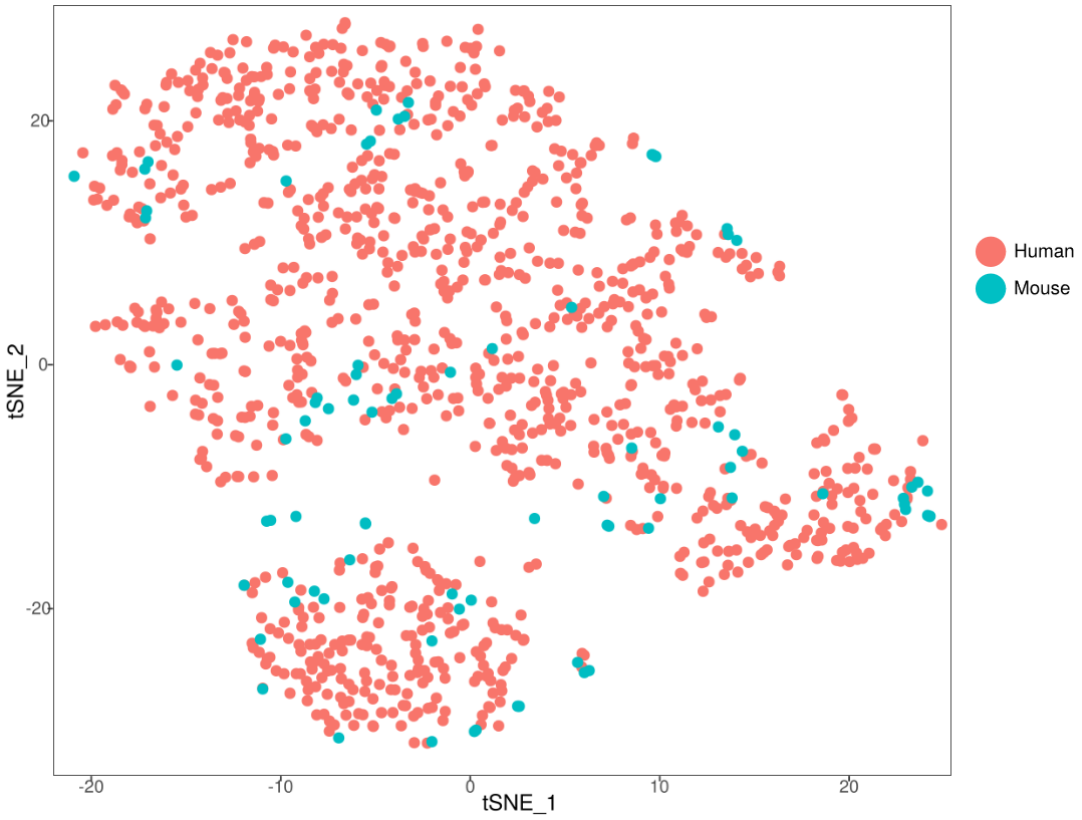

Figure 4.3: Human and mouse dataset integrated using Canonical Correlation Analysis visualized with dimensionality reduction method t-SNE.

of luminal A and luminal B subtypes overlap. Here, normal-like samples are grouped together, and basal-like samples as in hierarchical clustering are clearly grouped. It can be noticed that a portion of healthy samples is similar to normal-like and luminal A tumors, and a portion of HER2-enriched tumors groups with luminal A subtypes (Figure 4.6).

4.3.2. Genefu and Seurat subtyping of human tumors

To assess the accuracy of subgrouping human TCGA breast tumors using modified PAM50 subtyping algorithm from genefu, 10-fold cross-validation was performed and the accuracy of 82.41% was obtained (Table 4.1) by calculating the percentage of correctly classified intrinsic subtypes. The accuracy of subtyping with Seurat showed to be lower, 67.62% (Table 4.2) because a great portion of luminal A subtypes is spread

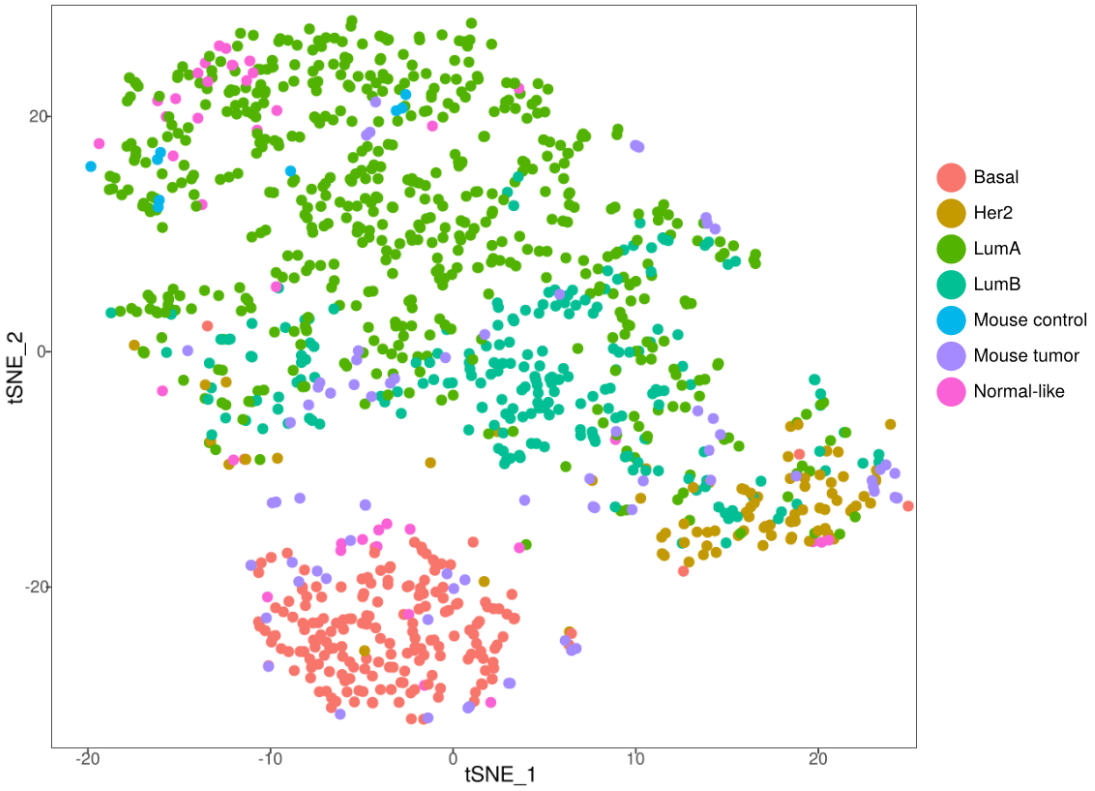

Figure 4.4: Human and mouse dataset integrated using Canonical Correlation Analysis visualized with dimensionality reduction method t-SNE. Human breast tumors are colored by the intrinsic subtype and mouse tumors are subtyped in relation to human tumors by assigning them to the nearest cluster.

among 3 different clusters, one of them being cluster that represents luminal B subtype. Basal-like, HER2-enriched subtypes, and healthy samples are clustered very accurately, but part of luminal A subtypes here is also grouped as luminal B.

4.3.3. Genefu and Seurat subtyping of mouse tumors

Results of subtyping using two different approaches are presented in Table **??**. Subtyping with Seurat resulted with each mouse tumor sample subtyping as one of the five intrinsic subtypes or as a healthy breast tissue. Genefu assigned one of the five intrinsic subtypes to each mouse sample and additionally calculated the probability to belong to each subtype. Mouse control samples, which are samples of the healthy breast tissue, are mostly subtyped as luminal A using both packages. It can be noticed that almost all

Table 4.1: The accuracy of genefu subtyping assessed using 10-fold cross-validation is 82.41%. TCGA breast tumors were subtyped using modified PAM50 subtyping algorithm from genefu package and the results were compared to clinically subtyped samples. True subtypes are column-wise, calculated subtypes are row-wise.

	Basal	**HER2**	**LumA**	**LumB**	**Normal**
Basal	**23**	1			
HER2		**6**			
LumA		6	**35**	6	7
LumB		3		**21**	
Normal	2				**4**

Table 4.2: Seurat classification of human RNA sequenced TCGA breast tumor samples and healthy breast tissue. Obtained accuracy is 67.62%. True subtypes are column-wise, calculated clusters representing each subtype are row-wise.

	Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**	**Cluster 6**
Basal	**184**		2		2	
Healthy	1	**107**		5		
HER2			**80**		2	
LumA		36	11	**245**	170	94
LumB	0	0	16	0	**186**	5
Normal	3	11	4	17	5	**0**

samples subtyped as luminal A have more than 30% chance to belong to normal-like subtype. A portion of samples that are tumor organoids (GSM2284739, GSM2284743, GSM2284741, GSM2284738, GSM2284742, GSM2284740) were subtyped as healthy breast tissue with Seurat, and those samples genefu subtyped the same as control samples. There are 39 samples that were subtyped the same with both packages, which is 52% when excluding samples subtyped as healthy using Seurat to make the two approaches comparable. Many tumor samples showed the same pattern in values of probabilities to belong to each subtype: they all have probabilities to belong to basal-like, HER2-enriched, and luminal B subtype, but no probability to belong to luminal A or normal-like subtype.

Table 4.3: Results of subtyping RNA sequenced mouse breast tumors in relation to human tumors by adapting PAM50 gene expression signature in R packages genefu and Seurat are presented in Seurat and genefu columns. Rows colored red are representing samples with disagreements between two subtyping approaches where one approach subtypes sample as either luminal A, healthy normal-like, and other approach subtypes sample as basal-like, HER2-enriched, or luminal B. Probabilities of mouse breast tumor samples to belong to each subtype calculated with PAM50 molecular subtyping algorithm from genefu are colored green with higher probabilities having higher color intensity. True subtypes are column-vise, calculated subtypes are row-vise.

Sample	Seurat	genefu	Basal	HER2	LumA	LumB	Normal
Control_1	LumA	Normal	0.00	0.00	0.47	0.00	0.53
Control_2	LumA	LumA	0.00	0.00	0.53	0.00	0.47
Control_3	LumA	Normal	0.00	0.00	0.50	0.00	0.50
Control_4	LumA	LumA	0.00	0.00	0.56	0.00	0.44
Control_5	LumA	LumA	0.00	0.00	0.54	0.00	0.46
Control_6	LumA	LumA	0.00	0.00	0.50	0.00	0.50
Control_7	LumA	LumA	0.00	0.00	0.57	0.00	0.43
Control_8	LumA	LumA	0.00	0.00	0.53	0.00	0.47

Table 4.3 continued from previous page

Control_9	LumA	LumA	0.00	0.00	0.54	0.00	0.46
GSM2284739	Healthy	Normal	0.00	0.00	0.46	0.00	0.54
GSM2284743	Healthy	Normal	0.00	0.00	0.42	0.00	0.58
GSM2284741	Healthy	Normal	0.00	0.00	0.20	0.00	0.80
GSM2284738	Healthy	Normal	0.00	0.00	0.41	0.00	0.59
GSM2284742	Healthy	Normal	0.00	0.00	0.40	0.00	0.60
GSM2284740	Healthy	Normal	0.00	0.00	0.48	0.00	0.52
GSM2098346	Basal	Normal	0.00	0.11	0.35	0.00	0.54
GSM2098345	Basal	Normal	0.00	0.00	0.00	0.00	1.00
GSM2098347	Basal	Her2	0.15	0.53	0.00	0.32	0.00
GSM2098348	Basal	Her2	0.33	0.35	0.00	0.31	0.00
GSM2178239	Basal	LumB	0.35	0.28	0.00	0.37	0.00
GSM2178240	Basal	LumB	0.34	0.28	0.00	0.38	0.00
GSM2178241	Basal	LumB	0.36	0.24	0.00	0.40	0.00
GSM2370617	LumB	LumB	0.36	0.24	0.00	0.40	0.00
GSM2044416	LumA	LumA	0.00	0.00	0.89	0.11	0.00
GSM2044417	LumA	LumA	0.03	0.00	0.53	0.00	0.44
GSM3057406	Her2	LumB	0.10	0.44	0.00	0.45	0.00
GSM3057407	Her2	Her2	0.18	0.41	0.00	0.40	0.00
GSM3057408	Her2	LumB	0.12	0.40	0.00	0.48	0.00
GSM3057409	Her2	LumB	0.11	0.44	0.00	0.45	0.00
GSM3057410	Her2	LumA	0.00	0.00	0.58	0.00	0.42
GSM3057411	Her2	Her2	0.06	0.47	0.00	0.47	0.00
GSM3057412	Basal	LumB	0.27	0.36	0.00	0.37	0.00
GSM3057413	Basal	LumB	0.27	0.36	0.00	0.38	0.00
GSM3057414	Basal	LumB	0.28	0.34	0.00	0.38	0.00
tumor_HL01	LumB	LumB	0.16	0.00	0.18	0.67	0.00
tumor_HL08	LumB	LumB	0.22	0.24	0.00	0.54	0.00

Table 4.3 continued from previous page

tumor_HL100	LumB	LumB	0.36	0.19	0.00	0.46	0.00
tumor_HL107	LumA	LumA	0.00	0.00	0.66	0.00	0.34
tumor_HL109	Basal	LumB	0.41	0.10	0.00	0.50	0.00
tumor_HL116	LumA	LumA	0.00	0.00	0.52	0.00	0.48
tumor_HL117	LumA	LumA	0.00	0.00	0.50	0.00	0.50
tumor_HL119	Basal	Basal	0.50	0.24	0.00	0.27	0.00
tumor_HL120	Healthy	LumA	0.00	0.00	0.53	0.00	0.47
tumor_HL121	LumB	LumB	0.25	0.25	0.00	0.50	0.00
tumor_HL123	LumB	Basal	0.43	0.31	0.00	0.26	0.00
tumor_HL124	LumB	LumB	0.33	0.28	0.00	0.39	0.00
tumor_HL125	LumA	Basal	0.42	0.30	0.00	0.28	0.00
tumor_HL126	LumA	Normal	0.22	0.00	0.26	0.00	0.52
tumor_HL127	LumB	Basal	0.54	0.28	0.00	0.19	0.00
tumor_HL128	Basal	Basal	0.75	0.15	0.00	0.10	0.00
tumor_HL130	Basal	Basal	0.66	0.00	0.00	0.00	0.34
tumor_HL132	LumB	LumB	0.31	0.27	0.00	0.42	0.00
tumor_HL133	Basal	Normal	0.31	0.20	0.00	0.00	0.49
tumor_HL134	Basal	Basal	0.46	0.31	0.00	0.23	0.00
tumor_HL135	Basal	Basal	0.57	0.26	0.00	0.17	0.00
tumor_HL136	Basal	Basal	0.90	0.00	0.00	0.00	0.10
tumor_HL137	LumB	LumB	0.25	0.00	0.00	0.75	0.00
tumor_HL145	LumB	Her2	0.27	0.37	0.00	0.36	0.00
tumor_HL147	LumB	Her2	0.19	0.40	0.08	0.33	0.00
tumor_HL149	Basal	Basal	0.47	0.26	0.00	0.27	0.00
tumor_HL151	Her2	LumA	0.00	0.00	0.69	0.00	0.31
tumor_HL152	LumA	Her2	0.19	0.58	0.00	0.00	0.23
tumor_HL156	Basal	Normal	0.31	0.19	0.00	0.00	0.50
tumor_HL157	LumB	LumA	0.00	0.00	0.85	0.15	0.00

Table 4.3 continued from previous page

tumor_HL158	LumB	LumA	0.00	0.00	0.91	0.09	0.00
tumor_HL160	LumB	LumB	0.00	0.00	0.40	0.60	0.00
tumor_HL161	Basal	LumB	0.30	0.29	0.00	0.41	0.00
tumor_HL162	LumB	LumA	0.00	0.00	0.57	0.43	0.00
tumor_HL163	Her2	Her2	0.32	0.37	0.00	0.32	0.00
tumor_HL166	LumB	LumB	0.15	0.20	0.00	0.65	0.00
tumor_HL.167.2	LumB	LumB	0.30	0.23	0.00	0.47	0.00
tumor_HL169	LumB	Basal	0.64	0.06	0.00	0.30	0.00
tumor_HL170	Basal	LumB	0.34	0.25	0.00	0.41	0.00
tumor_HL17	LumB	LumB	0.33	0.24	0.00	0.43	0.00
tumor_HL23	LumB	LumB	0.16	0.23	0.00	0.61	0.00
tumor_HL40	Basal	LumB	0.31	0.30	0.00	0.39	0.00
tumor_HL47	LumB	LumB	0.23	0.26	0.00	0.51	0.00
tumor_HL65	LumA	LumB	0.37	0.19	0.00	0.45	0.00
tumor_HL70	Basal	Basal	0.49	0.23	0.00	0.28	0.00
tumor_HL76	LumB	LumB	0.25	0.31	0.00	0.44	0.00
tumor_HL77	Basal	Basal	0.50	0.26	0.00	0.24	0.00
tumor_HL80	Basal	Normal	0.29	0.00	0.00	0.00	0.71

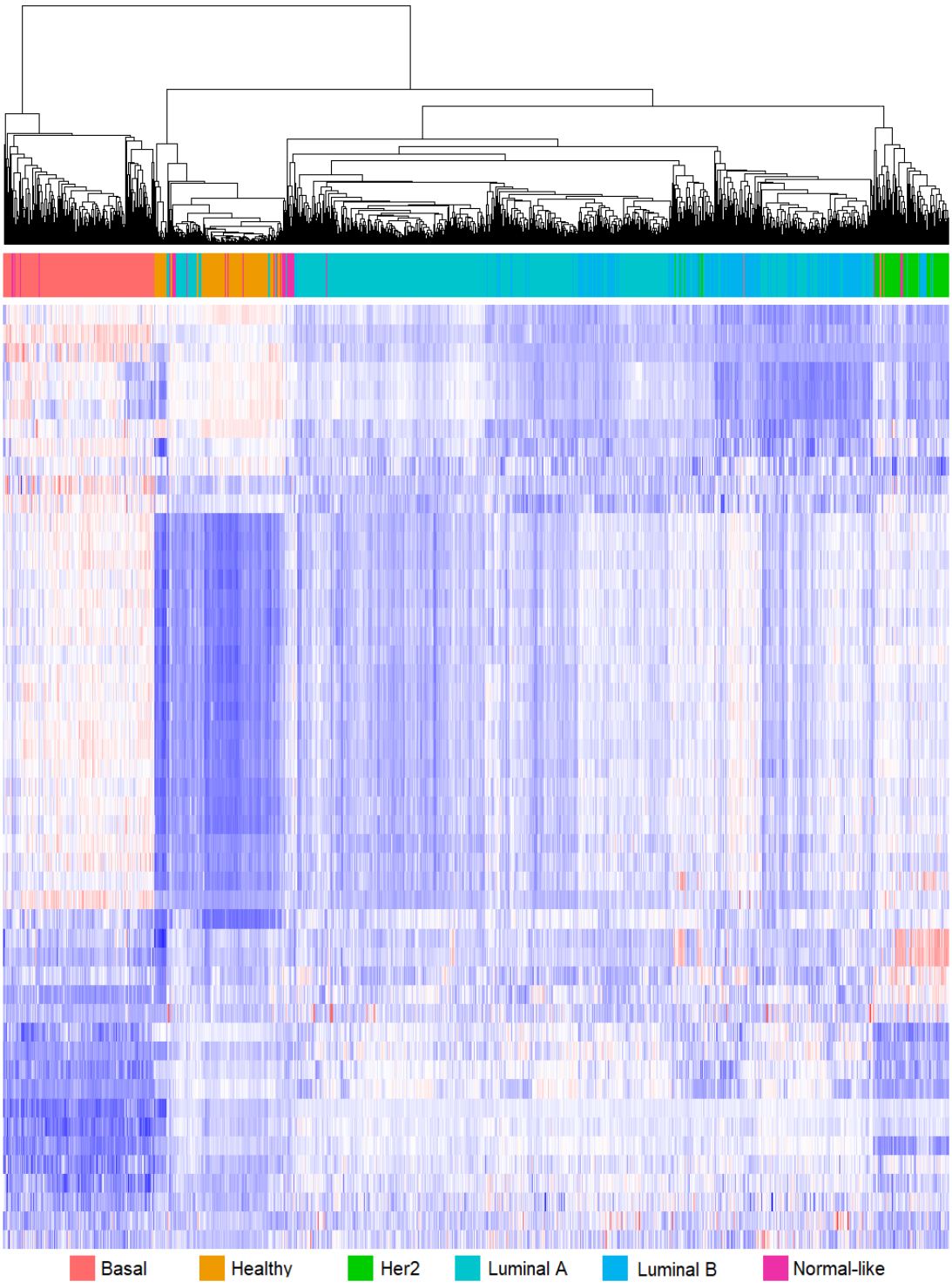

Figure 4.5: Hierarchical clustering of TCGA breast tumor samples performed using average linkage and Pearson's distance on PAM50 genes. Color coded label representing clinically determined intrinsic subtypes is added. High levels of expression are in red, low levels of expression are in blue.

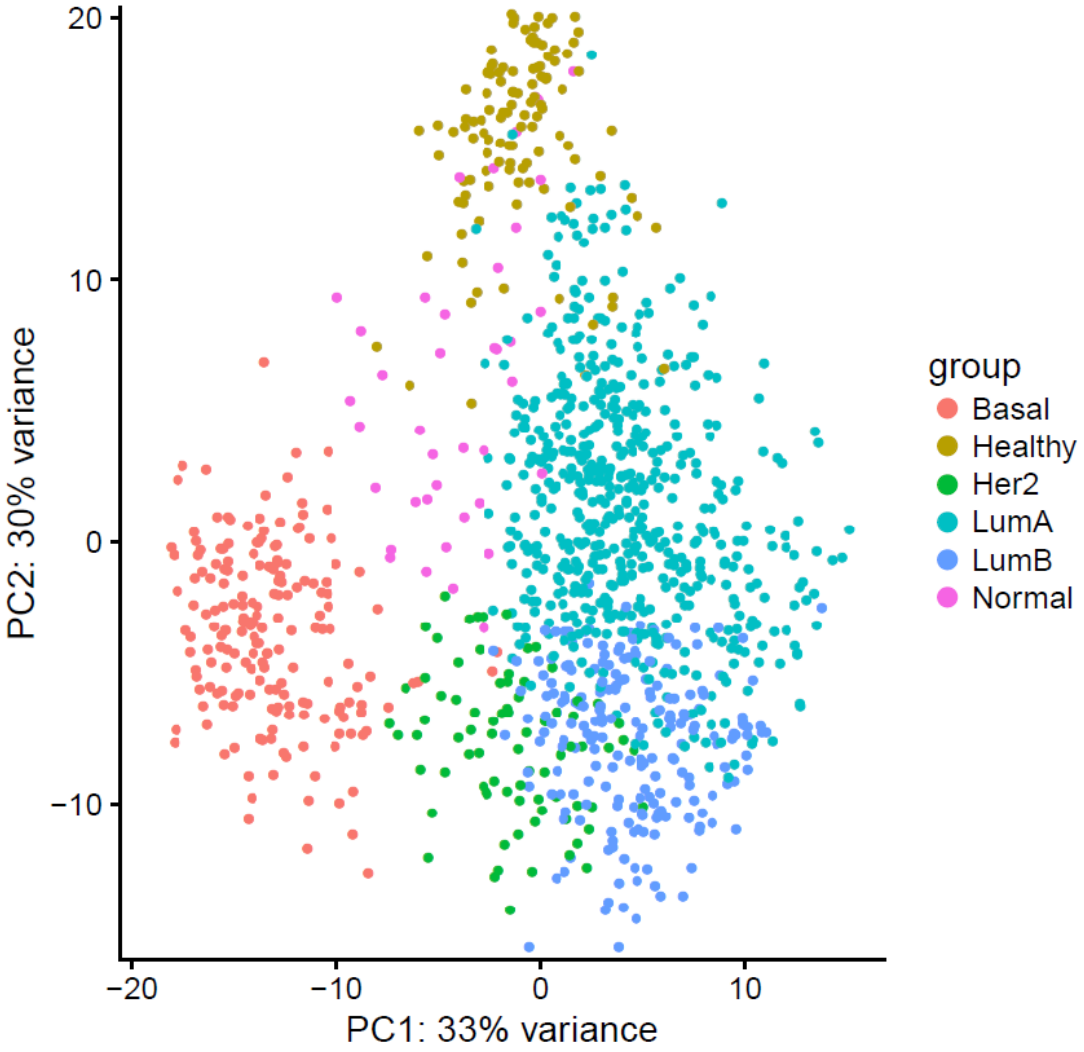

Figure 4.6: Principal Component Analysis of TCGA breast tumor samples calculated based on PAM50 genes. Points representing samples are colored according to clinically determined intrinsic subtypes.

5. Discussion

Heterogeneity of breast cancer that occurs at the morphological, genomic, transcriptomic and proteomic levels, creates challenges in diagnostics and limits the efficacy of breast cancer therapy (Turashvili and Brogi, 2017). To simplify the study of the molecular complexity of breast cancer, mouse tumor models are used, but the extent to which they model human breast cancer and are reflective of the human heterogeneity has yet to be demonstrated with gene expression studies on a large scale (Hollern and Andrechek, 2014). In this research, it was studied how mouse breast tumors reflect human intrinsic subtypes.

The first challenge was building mouse dataset by combining samples from many different sources and finding appropriate variables that would correct the batch effect and preserve biological differences among samples (Goh et al., 2017). We ended up with 82 samples from 6 different sources which was sufficient for the analysis. All non-protein coding genes and non-orthologs were discarded from both datasets. It was crucial to end up with the same number of ortholog genes in both species, where each gene has its corresponding pair in other species so that human and mouse datasets can be correctly merged. Although most orthologs were one-to-one, meaning the entry has only one ortholog in the other species, there were also one-to-many, many-to-one, and many-to-many orthologs which were selected according to the highest percentage of homology. Important step prior calculating new PAM50 centroids, from TCGA human breast cancer data, was to merge human and mouse datasets and correct them for organism type. Doing that, we accounted for possible differences in gene expression due to different physiology between species and make possible to subtype mouse

breast tumors in relation to human tumors. In contrast to merging datasets and calculating PAM50 centroids for modification of PAM50 subtyping algorithm, integration of two different datasets in Seurat did not require previous correction for batch effects because CCA integration method takes technical differences into account and calculates correction vectors. In this research, VST method for transforming expression data was applied. When using Seurat for RNA-Seq data analysis instead of single-cell RNA-Seq data, it is important to make sure that "normalization.method" parameter in CreateSeuratObject() function is set to NULL.

It was noticed that some subtypes show more similarity to the others. In 10-fold cross-validation, although it showed 82.41% accuracy, genefu subtyped few samples as normal-like, while they were actually luminal A subtype. Luminal A and normal-like subtypes have the most favorable prognosis among all intrinsic subtypes (Toft and Cryns, 2010). Additionally, normal-like has gene expression pattern similar to the ones found in normal breast tissue samples. Hierarchical clustering and principal component analysis of TCGA breast cancer dataset, showed that portion of luminal A groups with luminal B subtypes, while basal tumors were clearly separated. The similarity between a portion of luminal A and luminal B tumors can be explained by their gene expression pattern similar to the luminal epithelium of the mammary gland which includes genes such as the estrogen receptor (ER) and progesterone receptor (PR). They show differences in the expression of HER2 gene, where luminal A subtype is HER2 negative, and luminal B subtype is HER2 positive (Vallejos et al., 2010). This HER2 positive characteristic of luminal B subtype can also be noticed while looking at the results of subtyping (Table **??**) where tumors subtyped as luminal B always show some probability to belong to HER2-enriched subtype.

Seurat based subtyping identified all 9 control samples as luminal A, while genefu subtyped 2 of them as normal-like and the rest as luminal A subtype. There were 6 samples downloaded from ARCHS4 that were organoids of breast tumor which subtyped as healthy breast samples according to Seurat, and as normal-like tumors according to Seurat. Almost all mouse breast tumors subtype as either luminal B, HER2-enriched or

basal-like tumors, and have some probability to belong to all three mentioned subtypes. These are intrinsic subtypes that have worse prognosis than luminal A or normal-like subtypes (Fan et al., 2006).

Since congruence between intrinsic subtypes of mouse breast tumors determined with genefu and Seurat is only 52%, we can not draw conclusions about specific subtype of particular mouse tumor, but we can confidently distinguish between tumor and control samples. Both approaches showed the agreement between subtyping non-tumors as either luminal A or normal-like subtype, and tumors as either luminal B, HER2-enriched or basal-like. The advantage of PAM50 molecular subtyping algorithm from genefu is calculating the probability of each sample to belong to specific subtype which can give us an additional information about the tumor heterogeneity. Using CCA implemented in Seurat is easier and faster approach since it does not require prior manipulation of the data such as batch correction. Additionally, it enables healthy samples to be included in the analysis whereas genefu restricts only to five intrinsic subtypes. The limitations of this approach include mouse samples being combined from many different sources and the need for their batch correction. The process of batch correction can unwantedly remove some of the important biological differences between the samples. This could be avoided if all mouse samples were sequenced within the same RNA-Seq experiment. Another limitation is the low number of mouse samples (82) in comparison to human samples (1186). By increasing the number of mouse breast tumor samples, more confident results can be obtained.

6. Conclusion

Based on this research, the following can be concluded:

- PAM50 molecular subtype classification algorithm implemented in R package genefu can be used for the analysis of RNA-Seq data.
- PAM50 molecular subtype classification algorithm implemented in R package genefu can be used to subtype mouse breast tumors in relation of human tumors after performing transformation and batch correction for organism type.
- Canonical correlation analysis implemented in R package Seurat can be used to integrate human and mouse bulk RNA-Seq data.
- Canonical correlation analysis implemented in R package Seurat can be used to subtype mouse breast tumors in relation to human breast tumors.
- Mouse control samples with both methods subtype as luminal A with probability to belong to normal-like subtype.
- Mouse breast tumor samples most often subtype as HER2-enriched, basal-like, and luminal B subtype.
- The use of more mouse breast tumor samples from the same batch would contribute to better assessment of the two used approaches for breast tumor subtyping.

REFERENCES

Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411, 2018.

Fatima Cardoso, Laura J van't Veer, Jan Bogaerts, Leen Slaets, Giuseppe Viale, Suzette Delaloge, Jean-Yves Pierga, Etienne Brain, Sylvain Causeret, Mauro DeLorenzi, et al. 70-gene signature as an aid to treatment decisions in early-stage breast cancer. *New England Journal of Medicine*, 375(8):717–729, 2016.

Antonio Colaprico, Tiago C Silva, Catharina Olsen, Luciano Garofano, Claudia Cava, Davide Garolini, Thais S Sabedot, Tathiane M Malta, Stefano M Pagnotta, Isabella Castiglioni, et al. Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. *Nucleic acids research*, 44(8):e71–e71, 2015.

Marco Colleoni, Zhuoxin Sun, Karen N Price, Per Karlsson, John F Forbes, Beat Thürlimann, Lorenzo Gianni, Monica Castiglione, Richard D Gelber, Alan S Coates, et al. Annual hazard rates of recurrence for breast cancer during 24 years of follow-up: results from the international breast cancer study group trials i to v. *Journal of Clinical Oncology*, 34(9):927, 2016.

Sylvain Delaunay, Francesca Rapino, Lars Tharun, Zhaoli Zhou, Lukas Heukamp, Martin Termathe, Kateryna Shostak, Iva Klevernic, Alexandra Florin, Hadrien Desmecht, et al. Elp3 links trna modification to ires-dependent translation of lef1 to sustain metastasis in breast cancer. *Journal of Experimental Medicine*, 213(11): 2503–2523, 2016.

Cheng Fan, Daniel S Oh, Lodewyk Wessels, Britta Weigelt, Dimitry SA Nuyten, Andrew B Nobel, Laura J van't Veer, and Charles M Perou. Concordance among gene-expression–based predictors for breast cancer. *New England Journal of Medicine*, 355(6):560–569, 2006.

Debora Fumagalli, Alexis Blanchet-Cohen, David Brown, Christine Desmedt, David Gacquer, Stefan Michiels, Françoise Rothé, Samira Majjaj, Roberto Salgado, Denis Larsimont, et al. Transfer of clinically relevant gene expression signatures in breast cancer: from affymetrix microarray to illumina rna-sequencing technology. *BMC genomics*, 15(1):1008, 2014.

Deena MA Gendoo, Natchar Ratanasirigulchai, Markus S Schröder, Laia Paré, Joel S Parker, Aleix Prat, and Benjamin Haibe-Kains. Genefu: an r/bioconductor package for computation of gene expression-based signatures in breast cancer. *Bioinformatics*, 32(7):1097–1099, 2015.

Wilson Wen Bin Goh, Wei Wang, and Limsoon Wong. Why batch effects matter in omics data, and how to avoid them. *Trends in biotechnology*, 35(6):498–507, 2017.

Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. Batch effects in single-cell rna-sequencing data are corrected by matching mutual nearest neighbors. *Nature biotechnology*, 36(5):421, 2018.

Lyndsay N Harris, Nofisat Ismaila, Lisa M McShane, Fabrice Andre, Deborah E Collyar, Ana M Gonzalez-Angulo, Elizabeth H Hammond, Nicole M Kuderer, Minetta C Liu, Robert G Mennel, et al. Use of biomarkers to guide decisions on adjuvant systemic therapy for women with early-stage invasive breast cancer: American society of clinical oncology clinical practice guideline. *Journal of Clinical Oncology*, 34(10):1134, 2016.

Stephanie C Hicks and Rafael A Irizarry. Quantro: a data-driven approach to guide the choice of an appropriate normalization method. *Genome biology*, 16(1):117, 2015.

Daniel P Hollern and Eran R Andrechek. A genomic analysis of mouse models of breast cancer reveals molecular features ofmouse models and relationships to human breast cancer. *Breast Cancer Research*, 16(3):R59, 2014.

Kimberly R Kukurba and Stephen B Montgomery. Rna sequencing and analysis. *Cold Spring Harbor Protocols*, 2015(11):pdb–top084970, 2015.

Alexander Lachmann, Denis Torre, Alexandra B Keenan, Kathleen M Jagodnik, Hoyjin J Lee, Lily Wang, Moshe C Silverstein, and Avi Ma'ayan. Massive mining of publicly available rna-seq data from human and mouse. *Nature communications*, 9(1):1366, 2018.

Ling Liu and M Tamer Özsu. *Encyclopedia of database systems*, volume 6. Springer New York, NY, USA:, 2009.

Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550, 2014.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Cancer Genome Atlas Network et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61, 2012.

Alicia Oshlack and Matthew J Wakefield. Transcript length bias in rna-seq data confounds systems biology. *Biology direct*, 4(1):14, 2009.

Mi Kyung Park, Chang Hoon Lee, and Ho Lee. Mouse models of breast cancer in preclinical research. *Laboratory animal research*, 34(4):160–165, 2018.

Taesung Park, Sung-Gon Yi, Sung-Hyun Kang, SeungYeoun Lee, Yong-Sung Lee, and Richard Simon. Evaluation of normalization methods for microarray data. *BMC bioinformatics*, 4(1):33, 2003.

Joel S Parker, Michael Mullins, Maggie CU Cheang, Samuel Leung, David Voduc, Tammi Vickery, Sherri Davies, Christiane Fauron, Xiaping He, Zhiyuan Hu, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. *Journal of clinical oncology*, 27(8):1160, 2009.

Charles M Perou, Therese Sørlie, Michael B Eisen, Matt Van De Rijn, Stefanie S Jeffrey, Christian A Rees, Jonathan R Pollack, Douglas T Ross, Hilde Johnsen, Lars A Akslen, et al. Molecular portraits of human breast tumours. *nature*, 406(6797):747, 2000.

Charles Schmidt. Mammaprint reveals who can skip chemotherapy for breast cancer. *JNCI: Journal of the National Cancer Institute*, 108(8), 2016.

Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2016. *CA: a cancer journal for clinicians*, 66(1):7–30, 2016.

Therese Sørlie, Charles M Perou, Robert Tibshirani, Turid Aas, Stephanie Geisler, Hilde Johnsen, Trevor Hastie, Michael B Eisen, Matt Van De Rijn, Stefanie S Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98 (19):10869–10874, 2001.

Therese Sørlie, Robert Tibshirani, Joel Parker, Trevor Hastie, James Stephen Marron, Andrew Nobel, Shibing Deng, Hilde Johnsen, Robert Pesich, Stephanie Geisler, et al. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proceedings of the national academy of sciences*, 100(14):8418–8423, 2003.

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 2019.

Daniel J Toft and Vincent L Cryns. Minireview: basal-like breast cancer: from molecular profiles to targeted therapies. *Endocrine Reviews*, 31(5):776–777, 2010.

Gulisa Turashvili and Edi Brogi. Tumor heterogeneity in breast cancer. *Frontiers in medicine*, 4:227, 2017.

Carlos S Vallejos, Henry L Gómez, Wilder R Cruz, Joseph A Pinto, Richard R Dyer, Raúl Velarde, Juan F Suazo, Silvia P Neciosup, Mauricio León, A Miguel, et al. Breast cancer classification according to immunohistochemistry markers: subtypes and association with clinicopathologic variables in a peruvian hospital database. *Clinical breast cancer*, 10(4):294–300, 2010.

Ludo Waltman and Nees Jan Van Eck. A smart local moving algorithm for large-scale modularity-based community detection. *The European Physical Journal B*, 86(11): 471, 2013.

Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57, 2009.

Appendix A

Data preparation for mouse RNA-Seq data analysis

Ivna Ivanković

Load required packages, set working directory.

```
suppressMessages( library(DESeq2) )
suppressMessages( library(Seurat) )
suppressMessages( library(genefu) )
suppressMessages( library(biomaRt) )
suppressMessages( library(ggplot2) )
suppressMessages( library(dplyr) )
suppressMessages( library(data.table) )
suppressMessages( library(preprocessCore) )
suppressMessages( library(dendextend) )
suppressMessages( library(sva) )

setwd("home/R/project")
```

1. Mouse Data

Mouse dataset is prepared by combining samples from three different sources: raw counts of RNA sequenced mouse tumors available at Sequence Read Archive (SRA) with identifier SRP115453, RNA sequenced healthy breast tissue from DKFZ, and mouse breast tumor samples from ARCHS4 (Lachmann et al., 2018) database.

```
# raw counts available at Sequence Read Archive (SRA) with identifier
SRP115453
jonkers <- read.table("jonkers_count.tsv", header = T)

# dkfz controls, healthy breast tissue
dkfz_controls <- read.table("dkfz_controls.tsv", header = T)

# merge mouse tumor samples and controls
mouse_raw_counts <- cbind(jonkers, dkfz_controls)

save(mouse_raw_counts, file = "mouse_raw_counts.RData")
```

1.1. Download breast tumor samples from ARCHS4 database

The idea is to add more mouse samples for CCA Seurat analysis. I searched for mouse RNA seq data, ideally for breast tissue. There is a paper about **ARCHS4 database** where I found 47 mouse RNA seq breast tissue samples.

ARCHS4 is a web resource that makes the majority of published RNA-seq data from human and mouse available at the gene and transcript levels.

After the search is complete, the samples are highlighted and an auto-generated R script is provided for downloading the set of highlighted samples. Executing the R script builds a local expression matrix in tab-separated values format with the samples as columns and the genes as the rows.

Auto-generated R script

Retrieval date 13th February 2019.

I will only use one batch of mouse data. Library type for **GSE81380 batch is polyA** according to the information obtained from auto generated R script: *library = h5read(destination_file, "meta/Sample_extract_protocol_ch1")*

```
setwd("/icgc/dkfzlsdf/analysis/B060/Breast_TCGA_ivna/data")

# R script to download selected samples
# Copy code and run on a local machine to initiate download

# Check for dependencies and install if missing
packages <- c("rhdf5")
if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
    print("Install required packages")
    source("https://bioconductor.org/biocLite.R")
    biocLite("rhdf5")
}
library("rhdf5")
library("tools")

destination_file = "mouse_matrix_download.h5"
extracted_expression_file = "Breast_expression_matrix.tsv"
url = "https://s3.amazonaws.com/mssm-seq-matrix/mouse_matrix.h5"

# Check if gene expression file was already downloaded and check
integrity, if not in current directory download file form repository
if(!file.exists(destination_file)){
    print("Downloading compressed gene expression matrix.")
    download.file(url, destination_file, quiet = FALSE)
} else{
    print("Verifying file integrity...")
    checksum = md5sum(destination_file)

    if(destination_file == "human_matrix_download.h5"){
        # human checksum (checksum is for latest version of ARCHS4
```

```
data)
        correct_checksum = "f78da4a1855ff20da768eed1b73508be"
    } else{
        # mouse checksum (checksum is for latest version of ARCHS4
data)
        correct_checksum = "065abb20d2b9d2661e74328de8d23eb3"
    }

    if(checksum != correct_checksum){
        print("Existing file looks corrupted or is out of date.
Downloading compressed gene expression matrix again.")
        download.file(url, destination_file, quiet = FALSE)
    } else{
        print("Latest ARCHS4 file already exists.")
    }
}

checksum = md5sum(destination_file)
if(destination_file == "human_matrix_download.h5"){
    # human checksum (checksum is for latest version of ARCHS4 data)
    correct_checksum = "f78da4a1855ff20da768eed1b73508be"
} else{
    # mouse checksum (checksum is for latest version of ARCHS4 data)
    correct_checksum = "065abb20d2b9d2661e74328de8d23eb3"
}

if(checksum != correct_checksum){
    print("File download ran into problems. Please try to download
again. The files are also available for manual download at
http://amp.pharm.mssm.edu/archs4/download.html.")
} else{
    # Selected samples to be extracted
    samp =
c("GSM2284739","GSM2284743","GSM2284741","GSM2284738","GSM1013599","GSM
2284742","GSM2284740","GSM2098346","GSM2098345","GSM2098347","GSM209834
8","GSM1973811","GSM1973812","GSM2151462","GSM2151459","GSM2151453","GS
M2151467","GSM2151465","GSM2151455","GSM2151452","GSM2151454","GSM21514
56","GSM2151461","GSM2151460","GSM2151466","GSM2151464","GSM2151463","G
SM2151458","GSM2151457","GSM2178239","GSM2178240",
"GSM2178241","GSM2370617","GSM2044416","GSM2044417","GSM2044418","GSM30
16433","GSM3016434","GSM3057406","GSM3057407","GSM3057408","GSM3057409"
,"GSM3057410","GSM3057411","GSM3057412","GSM3057413","GSM3057414","")

    # Retrieve information from compressed data
    samples = h5read(destination_file, "meta/Sample_geo_accession")
    tissue = h5read(destination_file, "meta/Sample_source_name_ch1")
    genes = h5read(destination_file, "meta/genes")
    series = h5read(destination_file, "meta/Sample_series_id")
    library = h5read(destination_file,
```

```
"meta/Sample_extract_protocol_ch1")
    a <- h5read(destination_file, "meta/Sample_extract_protocol_ch1")

    # Identify columns to be extracted
    sample_locations = which(samples %in% samp)

    # extract gene expression from compressed data
    expression = h5read(destination_file, "data/expression",
index=list(1:length(genes), sample_locations))
    H5close()
    rownames(expression) = genes
    colnames(expression) = samples[sample_locations]
    series <- series[sample_locations]
    library <- library[sample_locations]

    # this is the batch I decided to use
    batch_samples <- samples[sample_locations][which(series ==
"GSE81380")]
    batch <- expression[, batch_samples]
    aa <- a[sample_locations][which(series == "GSE81380")]

    # Print file
    write.table(expression, file=extracted_expression_file, sep="\t",
quote=FALSE)
    print(paste0("Expression file was created at ", getwd(), "/",
extracted_expression_file))
}

setwd("/icgc/dkfzlsdf/analysis/B060/Breast_TCGA_ivna/")
```

In the auto-generated R script I added few lines to gain the information about library preparation and series.

```
series = h5read(destination_file, "meta/Sample_series_id")
library = h5read(destination_file, "meta/Sample_extract_protocol_ch1")
series <- series[sample_locations]
library <- library[sample_locations]
```

1.2. Identify outliers

Apply quantile normalization

```
# plot sample similarity
boxplot(log2(1+expression[,sample(1:ncol(expression), 16)]), main="read
count distribution by sample")

# here we apply quantile normalization that will force the expression
distribution to be the same for all samples
exp <- normalize.quantiles(log2(1+expression))
dimnames(exp) <- dimnames(expression)
```

In this case outlier is sample **GSM2044418** and it it removed from the dataset.

```
series <- c(rep("ctrl", 6), rep("archs4", 25), rep("tumors", 48))

# calculate pairwise correlation
cc <- cor(exp)
dend <- as.dendrogram(hclust(as.dist(1-cc)))
useries <- unique(series)
series_match <- useries[match(series, useries)]

# set colors to each series
colos <- colorspace::rainbow_hcl(length(useries), c = 160, l  = 50)
names(colos) = useries
series_color <- colos[series_match]

clu = cutree(dend, h=0.15)
labels_colors(dend) <- series_color[order.dendrogram(dend)]
dend <- color_branches(dend, h = 0.15)

par(mar = c(4,1,1,12))
plot(dend, horiz = TRUE)
colored_bars(cbind(clu, series_color), dend, rowLabels = c("Cluster",
"Series"), horiz = TRUE)
legend("topleft", legend = useries, fill = colos, bg="white", cex=0.6)

# subset largest cluster / drop outliers
largest_cluster = names(rev(sort(table(clu))))[1]
ww = which(clu == largest_cluster)
reduced_expression = exp[,ww]
reduced_series = series[ww]
```

After removal of the outlier sample, I will also remove series that consist of only one or two samples. These are:

- GSE41286; 1 sample
- GSE76075; 2 samples
- GSE110770; 2 samples

From 47 initial samples in 8 series, now there are 41 of them in 5 series.

It is also important to remove samples from batch **GSE8138** because those are actually human tumors transfered to mouse (mouse xenograft), so I might try adding those samples to human data and see how those classify. Those are HER2+ human breast tumors, harvested from SCID mice 2-days post treatment initiation, source I will remove those samples from the **expression** dataset.

```
# remove series that consist of only one or two samples
rser <- c("GSE41286", "GSE76075", "GSE110770")
series_filtered <- reduced_series[-c(which(reduced_series %in% rser))]
save(series_filtered, file = "series_filtered.RData")
```

```
# remove outlier
expression <- expression[, -which(colnames(expression) ==
"GSM2044418")]
expression <- expression[, -c(which(reduced_series %in% rser))]

# remove mouse xenografts
which(series_filtered == "GSE81380")
xenografts <- colnames(mouse_counts)[29:69][which(series_filtered ==
"GSE81380")]
```

1.3. Mapping mouse to human genes

BiomaRt using R. Webpage did not work good. According to ARCHS4 article supported genomes are Ensembl Homo sapiens **GRCh38** with the GRCh38.87 annotation file and Mus Musculus GRCm38 with the GRCm38.88 annotation file.

```
ensembl <- useMart(biomart = "ENSEMBL_MART_ENSEMBL",
                   path = "/biomart/martservice",
                   dataset = "mmusculus_gene_ensembl")

# creating query and downloading human and mouse gene names to do
mapping
output=getBM(attributes=c("ensembl_gene_id", "external_gene_name"),
       filters = "external_gene_name",
       values = rownames(expression),
       mart = ensembl)

expression <- data.frame(expression)
expression$external_gene_name <- rownames(expression)
exp_mouse <- merge(expression, output, by = "external_gene_name")
colnames(exp_mouse)[43] <- "GeneID"

# merge mouse data from all sources (controls, ARCHS4 and Jonkers
dataset)
mouse_raw_counts$GeneID <- rownames(mouse_raw_counts)
a <- merge(mouse_raw_counts, exp_mouse[,-1], by = "GeneID")
mouse_counts <- a[,-1]
rownames(mouse_counts) <- a[,1]

# save merged mouse data
save(mouse_counts, file = "mouse_counts.RData")
load(file = "mouse_counts.RData")
```

2. Human data

2.1. Download TCGA raw counts

Downloading raw RNAseq counts with **TCGAbiolinks**: An R/Bioconductor package for integrative analysis with GDC data. I downloaded tumor and normal samples. Read this paper.

```
if (!requireNamespace("BiocManager", quietly=TRUE))
    install.packages("BiocManager")
BiocManager::install("TCGAbiolinks")
library("TCGAbiolinks")

# preparing query and downloading samples
query <- GDCquery(project = "TCGA-BRCA",
                  data.category = "Transcriptome Profiling",
                  experimental.strategy = "RNA-Seq",
                  workflow.type = "HTSeq - Counts",
                  sample.type = c("Primary solid Tumor", "Solid Tissue
Normal"))

GDCdownload(query)

expdata <- GDCprepare(query)
expdata <- expdata[!duplicated(expdata)]
save(expdata, file = "expdata.RData")
load(file = "expdata.RData")

# get the sample information, here is also the information about PAM50
subtypes that I am interested in
sample.info <- SummarizedExperiment::colData(expdata)

# identify healthy samples
which(sample.info$definition == "Solid Tissue Normal")

# add information
sample.info$subtype_BRCA_Subtype_PAM50[which(sample.info$definition ==
"Solid Tissue Normal")] <- "Healthy"

# omit samples without information about PAM50 subtype
sample.info <- sample.info[-
c(which(is.na(sample.info$subtype_BRCA_Subtype_PAM50))), ]

save(sample.info, file = "sample.info.RData")
load(file = "sample.info.RData")
```

2.2. Identify outliers

```
# 1190 because it is the number of downloaded human samples
series <- rep("one", 1186)

# calculate pairwise correlation
cc <- cor(exp)
dend <- as.dendrogram(hclust(as.dist(1-cc)))
useries <- unique(series)
series_match <- useries[match(series, useries)]

# set colors to each series
colos <- colorspace::rainbow_hcl(length(useries), c = 160, l  = 50)
names(colos) = useries
series_color <- colos[series_match]

clu = cutree(dend, h=0.25)
labels_colors(dend) <- series_color[order.dendrogram(dend)]
dend <- color_branches(dend, h = 0.25)

par(mar = c(4,1,1,12))
plot(dend, horiz = TRUE)
colored_bars(cbind(clu, series_color), dend, rowLabels = c("Cluster",
"Series"), horiz = TRUE)
legend("topleft", legend = useries, fill = colos, bg="white", cex=0.6)

# subset largest cluster / drop outliers
largest_cluster = names(rev(sort(table(clu))))[1]
ww = which(clu == largest_cluster)
reduced_expression = exp[,ww]
reduced_series = series[ww]

# outlier detection
outlier_cluster = names(rev(sort(table(clu))))[2]
ww = which(clu == outlier_cluster)
outliers= colnames(exp[,ww])
```

There were no outliers in human TCGA dataset.

3. Human and mouse orthologs

```
ensembl <- useMart(biomart = "ENSEMBL_MART_ENSEMBL",
                   host = "grch37.ensembl.org",
                   path = "/biomart/martservice",
                   dataset = "hsapiens_gene_ensembl")

filters <- listFilters(ensembl)
attributes <- listAttributes(ensembl)
attributes[grep("mmusculus", attributes$name),]
```

```
output <- c("ensembl_gene_id",
            "mmusculus_homolog_ensembl_gene",
            "mmusculus_homolog_orthology_type",
            "mmusculus_homolog_perc_id", #%id. target Mouse gene
identical to query gene
            "mmusculus_homolog_perc_id_r1") #%id. query gene identical
to target Mouse gene

orthologs <- getBM(output,
                   filter = "with_mmusculus_homolog",
                   values = TRUE,
                   mart = ensembl)

save(orthologs, file = "./results/orthologs.RData")
load(file = "./results/orthologs.RData")

# choose orthologs which have highest % identity
orthologs <- as.data.table(orthologs)
a <- orthologs[orthologs[, .I[which.max(mmusculus_homolog_perc_id)],
by=mmusculus_homolog_ensembl_gene]$V1]
b <- a[a[, .I[which.max(mmusculus_homolog_perc_id_r1)],
by=ensembl_gene_id]$V1]

orthologs_unique <- data.frame(b[, c(1, 2)])

save(orthologs_unique, file = "./results/orthologs_unique.RData")
load(file = "./results/orthologs_unique.RData")

human_orthologs <- expdata[which(rownames(reduced_expdata) %in%
orthologs_unique$ensembl_gene_id),]
mouse_orthologs <- mouse_counts[which(rownames(mouse_counts) %in%
orthologs_unique$mmusculus_homolog_ensembl_gene),]

mouse_orthologs$mmusculus_homolog_ensembl_gene <-
rownames(mouse_orthologs)
mouse_orthologs <- merge(mouse_orthologs, orthologs_unique)
rownames(mouse_orthologs) <- mouse_orthologs$ensembl_gene_id
mouse_orthologs <- mouse_orthologs[, -71]

save(human_orthologs, file = "./results/human_orthologs.RData")
load(file = "./results/human_orthologs.RData")
save(mouse_orthologs, file = "./results/mouse_orthologs.RData")
load(file = "./results/mouse_orthologs.RData")

intersect_orthologs <- intersect(rownames(human_orthologs),
rownames(mouse_orthologs))
save(intersect_orthologs, file = "./results/intersect_orthologs.RData")

human_intersect_orthologs <- human_orthologs[intersect_orthologs,]
```

```
mouse_intersect_orthologs <- mouse_orthologs[intersect_orthologs,]

save(human_intersect_orthologs, file =
"./results/human_intersect_orthologs.RData")
save(mouse_intersect_orthologs, file =
"./results/mouse_intersect_orthologs.RData")

load(file = "./results/human_intersect_orthologs.RData")
load(file = "./results/mouse_intersect_orthologs.RData")
```

4. Variance Stabilizing Transformation

```
mvst <-
varianceStabilizingTransformation(as.matrix(mouse_intersect_orthologs[,
-1]), blind = TRUE, fitType = "parametric")
save(mvst, file = "mvst.RData")
load(file = "mvst.RData")

hvst <-
varianceStabilizingTransformation(assay(human_intersect_orthologs),
blind = TRUE, fitType = "parametric")
save(hvst, file = "hvst.RData")
load(file = "hvst.RData")
```

4. PAM50 genes

BiomaRt PAM50 annotation

According to supplementary information *For each sample, filter-passed reads were aligned to the **NCBI build 37 (hg19)** human reference sequence (GRCh37-lite) using BWA.*

To work with older reference assembly **grch37** I am using code from [this website] (https://davetang.org/muse/2012/04/27/learning-to-use-biomart/).

In **mart outputs** above, I am getting not perfect number of genes. In the **mart_output_name** file there are only 47 external_gene_name values. Therefore I decided to use **mart_output_id** with 51 external_gene_name values and now I will fix the annotation to have 50 unique values for ensembl_gene_id, external_gene_name and mmusculus_homolog_ensembl_gene.

At the end of the chunk below, I ordered mart_output_id according to external_gene_name and dropped the first row where external_gene_name was **AC217779.2** because this is not in PAM50 gene list.

mart_output_id file contains annotation for PAM50 genes column with:

- ensembl_gene_id

- external gene name (corresponds to name of PAM50 centroids)
- mmusculus_homolog_ensembl_gene

```
grch37 <- useMart(biomart="ENSEMBL_MART_ENSEMBL",
                  host="grch37.ensembl.org",
                  path="/biomart/martservice")

mart <- useMart(biomart="ENSEMBL_MART_ENSEMBL",
                host="grch37.ensembl.org",
                path="/biomart/martservice",
                dataset="hsapiens_gene_ensembl")

mart_output_id <- getBM(attributes=c("ensembl_gene_id",
"mmusculus_homolog_ensembl_gene", "external_gene_name"),
                      filters = "entrezgene",
                      values = pam50$centroids.map$EntrezGene.ID,
                      mart = mart)

mart_output_name <- getBM(attributes=c("ensembl_gene_id",
"mmusculus_homolog_ensembl_gene", "external_gene_name"),
                      filters = "external_gene_name",
                      values = rownames(pam50$centroids.map),
                      mart = mart)

mart_output_id <-
mart_output_id[order(mart_output_id$external_gene_name),]
mart_output_id <- mart_output_id[-1,]
```

I need to choose which mmusculus_homolog_ensembl_gene to drop. There are duplicates of MIA and triplicates of NAT1. The dropping was done according to the results of manual search of [MGI Mouse Vertebrate Homology database] (http://www.informatics.jax.org/homology.shtml).

Firstly I dropped mouse ENSMUSG00000095538 gene, because MIA ortholog is ENSMUSG00000089661.

In mouse there are 3 NAT1 homologs (Nat1. Nat2, Nat3) and the one with the highest variance (ENSMUSG00000051147, Nat2) is chosen for the further analysis. Therefore, rows 30 and 31 were also removed. At the end, the information about missing genes were manually added.

```
# drop mouse ENSMUSG00000095538 gene
rownames(mart_output_id) <- seq(1, 53)
mart_output_id <- mart_output_id[-32,]

# Remove version number in GeneID column
rownames(mouse_raw_counts) <- gsub("\\..*","", mouse_raw_counts$GeneID)

# remove control_1966 (alinment is not okay with this one)
mouse_raw_counts <- mouse_raw_counts[, -7]
```

```
# NAT1 duplicates, choosing one gene with highest variance
nat <-
mart_output_id[which(mart_output_id$external_gene_name=="NAT1"),"mmuscu
lus_homolog_ensembl_gene"]
var <- matrixStats::rowVars(as.matrix(mouse_raw_counts[nat,-1]))

# drop mouse ENSMUSG00000025588 and ENSMUSG00000056426 genes
rownames(mart_output_id) <- seq(1, 52)
mart_output_id <- mart_output_id[-c(37,38),]
rownames(mart_output_id) <- seq(1, 50)
```

Since gene synonyms exist, there is some inconsistency in naming. Therefore I replaced NUF2, NDC80 and ORC6 with their synonyms CDCA1, KNTC2 and ORC6L respectively and ordered **mart_output_id** file alphabetically according to external_gene_name.

```
mart_output_id$external_gene_name[which(mart_output_id$external_gene_na
me == "NUF2")] <- "CDCA1"
mart_output_id$external_gene_name[which(mart_output_id$external_gene_na
me == "NDC80")] <- "KNTC2"
mart_output_id$external_gene_name[which(mart_output_id$external_gene_na
me == "ORC6")] <- "ORC6L"

mart_output_id <-
mart_output_id[order(mart_output_id$external_gene_name),]
rownames(mart_output_id) <- seq(1:50)

identical(mart_output_id$external_gene_name, rownames(pam50$centroids))

# there are two possible human orthologs to mouse gene
ENSMUSG00000051147, here manually put ENSG00000156006 instead of
ENSG00000171428
mart_output_id[39,1] <- "ENSG00000156006"

save(mart_output_id, file = "./results/mart_output_id.RData")
load(file = "./results/mart_output_id.RData")
```

Appendix B

Genefu modification

Ivna Ivanković

Genefu package was modified to subtype RNA sequenced mouse breast tumors in relation to human tumors by adapting PAM50 gene expression signature.

It was modified to subtype RNA-Seq data instead of microarray data as an input by manually calculating PAM50 centroids specific to the set of downloaded human breast cancer samples from the TCGA. These centroids were used to subtype RNA sequenced mouse breast tumors in relation to intrinsic subtypes of human breast tumors.

1. Batch correction

Information on series according to GEO (Gene Expression Omnibus)

Organism	Batch	Strand Specifity	Library Type	Source
mouse	DKFZ	yes	ribo zero	unknown
mouse	GSE85810	no	poly A	GEO
mouse	GSE77107	yes	poly A	GEO, ScienceDirect
human	GSE81380 (HER2)	no	poly A	GEO
mouse	GSE81941	no	poly A	Oncotarget
mouse	GSE112094	no	poly A	GEO
human	TCGA	no	poly A	Suppl, Biostars

First, I will correct for batch effect in mouse dataset. Then, I will combine human and mouse datasets and correct them for organism type.

Mouse and human data is merged and corrected for organism type.

```
load(file = "hvst.RData")
load(file = "mvst.RData")

load(file = "mart_output_id.RData")
load(file = "series_filtered.RData")
series_filtered <- series_filtered[-c(which(series_filtered ==
"GSE81380"))]
load(file = "sample.info.RData")

mouse <- Seurat::CreateSeuratObject(raw.data = mvst, min.cells = 0,
```

```
min.genes = 0,
                              project = "Mouse_RNAseq",
                              normalization.method = NULL)

# add information about library type preparation method into meta.data
slot
librarytype <- c(rep("ribozero", 9), rep("polya", 25), rep("polya",
48))
mouse@meta.data$librarytype <- librarytype

# add information about strand specificity into meta.data slot
strandspecificity <- c(rep("non", 9), rep("non", 6), rep("specific",
4),
                       rep("non", 4), rep("specific", 2), rep("non", 9),
rep("non", 48))
mouse@meta.data$strandspecificity <- strandspecificity

mouse <- Seurat::ScaleData(object = mouse,
                    check.for.norm = FALSE,
                    vars.to.regress = "librarytype",
"strandspecificity",
                    model.use = "linear")

# merge human and mouse data, and correct for organism type
vst <- cbind(mouse@scale.data, hvst)

both <- Seurat::CreateSeuratObject(raw.data = vst, min.cells = 0,
min.genes = 0,
                              project = "Both_RNAseq",
                              normalization.method = NULL)

# add information about organism into meta.data slot
organism <- c(rep("mouse", 82), rep("human", 1186))
both@meta.data$organism <- organism

both <- Seurat::ScaleData(object = both,
                    check.for.norm = FALSE,
                    vars.to.regress = "organism",
                    model.use = "linear")
```

2. Calculate centroids

I am loading **both.RData** file which is Seurat object made in **data.preparation.Rmd** script. Short description: I used variance stabilizing transformation to transform mouse and human values separately. Then, transformed mouse values were corrected for library type and strand specificity using Seurat ScaleData function. Then, mouse and human data were merged and corrected for organism type again using Seurat ScaleData function.

```r
setwd("home/R/project")

# merged, VST, batch corrected human and mouse data
load(file = "both.RData")
# supplementary information for human data contain the information
about PAM50 subtypes
load(file = "sample.info.RData")
# PAM50 annotation obtained with biomaR
load(file = "mart_output_id.RData")
```

3. Cross-Validation

```r
# subsetting only human data according to indices from merged dataset
human.data <- both@scale.data[, 83:1268]
sample.info <- sample.info[colnames(trainData), ]

# subset only pam50 genes from the whole dataset
pam50.genes <- human.data[which(rownames(human.data) %in%
mart_output_id$ensembl_gene_id), ]
# choose only human tumor samples (remove healthy) that have
information about pam50 subtype in sample.info
pam50.val <- pam50.genes[,-
c(which(sample.info$subtype_BRCA_Subtype_PAM50 == "Healthy"))]


### 10-fold cross validation
yourdata <- pam50.val

#Randomly shuffle the data
yourdata <- yourdata[,sample(ncol(yourdata))]
#Create 10 equally size folds
folds <- cut(seq(1,ncol(yourdata)),breaks = 10,labels = FALSE)
#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds == i,arr.ind=TRUE)
  testData <- yourdata[, testIndexes]
  trainData <- yourdata[, -testIndexes]
}
```

4. Manually calculate centroids based on TCGA data

For each subtype centroids are calculated by averaging the expression values for PAM50 genes. I do that by grouping by subtypes, filtering each subtype and calculating means for each gene with colMeans function. Every subtype is stored in one data.frame, at the end I combine them and order genes according to alphabetical order of gene names. I save the results as **tcga.centroids**.

```r
# subsetting only human data according to indices from merged dataset
human.data <- both@scale.data[, 83:1268]

human.data <- trainData
sample.info <- sample.info[colnames(trainData), ]

# subset only pam50 genes from the whole dataset
pam50.genes <- human.data[which(rownames(human.data) %in%
mart_output_id$ensembl_gene_id), ]
# choose only human tumor samples (remove healthy) that have
information about pam50 subtype in sample.info
pam50.val <- pam50.genes[,-
c(which(sample.info$subtype_BRCA_Subtype_PAM50 == "Healthy"))]
# add column with subtype
pam50.sub <- cbind(t(pam50.val),
                   subtype = data.frame(subtype =
sample.info$subtype_BRCA_Subtype_PAM50[-
c(which(sample.info$subtype_BRCA_Subtype_PAM50 == "Healthy"))]))



basals <- pam50.sub  %>%
  group_by(subtype) %>%
  filter(subtype=="Basal")
basals <- colMeans(basals[,-51])

her2 <- pam50.sub %>%
  group_by(subtype) %>%
  filter(subtype=="Her2")
her2 <- colMeans(her2[,-51])

lumA <- pam50.sub %>%
  group_by(subtype) %>%
  filter(subtype=="LumA")
lumA <- colMeans(lumA[,-51])

lumB <- pam50.sub %>%
  group_by(subtype) %>%
  filter(subtype=="LumB")
lumB <- colMeans(lumB[,-51])

normal <- pam50.sub %>%
  group_by(subtype) %>%
  filter(subtype=="Normal")
normal <- colMeans(normal[,-51])

tcga.centroids <- t(rbind(Basal=basals,
                          Her2=her2,
                          LumA=lumA,
```

```
                          LumB=lumB,
                          Normal=normal))

# order calculated centroids according to order in mart_output_id
# this order is alphabetical regarding to gene names, the same as in
genefu pam50 dataset
tcga.centroids <- tcga.centroids[match(mart_output_id[,1],
rownames(tcga.centroids)),]

# and assign gene names to calculated centroids
# now they look like genefu pam50 centroids
rownames(tcga.centroids) <- mart_output_id[,3]

# and save them
save(tcga.centroids, file = "tcga.centroids.RData")
load(file = "tcga.centroids.RData")
```

5. Prepare mouse data

To use manually calculated centroids I am replacing default genefu centroids stored in **pam50.robust$centroids** with my manually calculated centroids **tcga.centroids**. That way I can normally use *molecular.subtyping* function.

```
# make variable with mouse data for subtyping
pam50.mouse <- both@scale.data[, 1:82][which(rownames(both@scale.data[,
1:82]) %in% mart_output_id$ensembl_gene_id), ]
```

Ordering genes to match the order of PAM50 centroids. This step is not necessary if I provide gene IDs in *annot* argument inside *molecular.subtyping* function. But I decided to go that way because I have all information in **mart_output_id** file.

```
pam50.mouse <- pam50.mouse[match(mart_output_id[,1],
rownames(pam50.mouse)),]
rownames(pam50.mouse) <- mart_output_id[,3]

annotation <- mart_output_id[, c(1,3)]
```

6. Genefu molecular subtyping

Apply genefu **intrinsic.cluster.predict** function to classify the subtypes according to manually calculated centroids from TCGA breast cancer RNAseq data. I transformed my mouse data because the function requires samples to be in rows and genes in columns.

```
# replace centroids from genefu with manually calculated centroids
based on TCGA RNA-Seq data
pam50.robust$centroids <- tcga.centroids
pam50$centroids <- tcga.centroids
```

```
# and subtype mouse tumors
preds <- intrinsic.cluster.predict(sbt.model=pam50,data=t(pam50.mouse),
annot=annotation,do.mapping=FALSE, do.prediction.strength=TRUE,
verbose=TRUE)

table(preds$subtype)
data.frame(preds$subtype)

# probabilities to belong to each subtype
preds$subtype.proba
```

Appendix C

Seurat modification

Ivna Ivanković

Seurat package was modified to integrate human and mouse bulk RNA sequenced data based on the set of PAM50 genes and used to determine intrinsic breast tumor subtypes.

1. Make Seurat object

Make separate human and mouse Seurat objects out of VST data with **CreateSeuratObject** function.

```
# load requierd packages
library(Seurat)
library(matrixStats)
library(dplyr)
library(DESeq2)
library(gplots)
library(tibble)

# set working directory
setwd("home/R/project")

# human samples, VST values for orthologs
load(file = "hvst.RData")
# information about PAM50 subtypes for human tumors
load(file = "sample.info.RData")

# mouse samples, VST values for orthologs
load(file = "mvst.RData")
# dataframe with PAM50 genes
load(file = "mart_output_id.RData")
# information about mouse samples downloaded from ARCHS4 database
load(file = "./results/series_filtered.RData")
# this series are mouse xenograft models, those are removed:
# since batch GSE81380 are HER2+ human breast tumors, harvested from
SCID mice 2-days post treatment initiation,
[source](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE81380) I
will remove those samples from mouse dataset
series_filtered <- series_filtered[-c(which(series_filtered ==
"GSE81380"))]
```

```
# remove healthy samples from human dataset
hvst <- hvst[, -c(which(sample.info$subtype_BRCA_Subtype_PAM50 ==
"Healthy"))]
sample.info <- sample.info[-
c(which(sample.info$subtype_BRCA_Subtype_PAM50 == "Healthy")),]


### CREATE SEURAT OBJECT
mouse <- CreateSeuratObject(raw.data = mvst, min.cells = 0, min.genes =
0,
                            project = "Mouse_RNAseq",
                            normalization.method = NULL)

human <- CreateSeuratObject(raw.data = hvst, min.cells = 0, min.genes =
0,
                            project = "Human_RNAseq",
                            normalization.method = NULL)

### MOUSE
mouse <- FindVariableGenes(object = mouse)
length(x = mouse@var.genes)

mouse <- ScaleData(object = mouse,
                   check.for.norm = FALSE,
                   model.use = "linear")

### HUMAN
human <- FindVariableGenes(object = human)
length(x = human@var.genes)

human <- ScaleData(object = human,
                   check.for.norm = FALSE,
                   model.use = "linear")
```

2. Merge human and mouse datasets

Human and mouse Seurat objects are merged in one seurat object called **merged** and Canonical Correlation Analysis is run with function RunCCA. Information about PAM50 molecular subtype for human data and phenotype information for mouse data are added to *@meta.data* slot of Seurat object.

```
# subset the list of PAM50 genes
pam50 <- mart_output_id[,1]

# gene selection
mouse_hvg <- rownames(x = head(x = mouse@hvg.info,  n = 10))
human_hvg <- rownames(x = head(x = human@hvg.info, n = 10))
hvg.union <- union(x = mouse_hvg, y = human_hvg)
```

```
# adding organism information into meta.data slot
human@meta.data[, "organism"] <- "Human"
mouse@meta.data[, "organism"] <- "Mouse"

# integration of human and mouse dataset based on PAM50 genes
merged <- RunCCA(object = human,
                 object2 = mouse,
                 genes.use = pam50)

# just change name
sample.info$subtype_BRCA_Subtype_PAM50[sample.info$subtype_BRCA_Subtype
_PAM50 == "Normal"] <- "Normal-like"

# adding information about human subtypes and mouse samples into
meta.data slot
merged@meta.data$phenoinfo <- c(sample.info$subtype_BRCA_Subtype_PAM50,
rep("Mouse control", 9), rep("Mouse tumor", 73))
```

3. Canonical Correlation Analysis

The good number of dimensions for my dataset is either 4 or 5. This is determined based on **DimHeatmap** plot. I was doing analysis with only PAM50 genes.

```
# visualize results of CCA plot CC1 versus CC2 and look at a violin
plot
p1 <- DimPlot(object = merged, reduction.use = "cca", group.by =
"organism", pt.size = 0.5,
              do.return = TRUE)
p2 <- VlnPlot(object = merged, features.plot = "CC1", group.by =
"organism",
              do.return = TRUE)
plot_grid(p1, p2)

# determine the number of dimensions to use in further analysis
PrintDim(object = merged, reduction.type = "cca", dims.print = 1:2,
genes.print = 10)

DimHeatmap(object = merged, reduction.type = "cca", cells.use = 500,
dim.use = 1:9,
           do.balanced = TRUE)

# now we align the CCA subspaces, which returns a new dimensional
reduction called cca.aligned
merged <- AlignSubspace(object = merged, reduction.type = "cca",
grouping.var = "organism",
                        dims.align = 1:6)

# visualize the aligned CCA and perform integrated analysis
p1 <- VlnPlot(object = merged, features.plot = "ACC1", group.by =
```

```
"organism",
              do.return = TRUE)
p2 <- VlnPlot(object = merged, features.plot = "ACC2", group.by =
"organism",
              do.return = TRUE)
plot_grid(p1, p2)

# now we can run a single integrated analysis on all cells
merged <- RunTSNE(object = merged, reduction.use = "cca.aligned",
dims.use = 1:6,
                do.fast = TRUE)

# SNN clustering
merged <- FindClusters(object = merged, reduction.type = "cca.aligned",
dims.use = 1:6,
                       resolution = 0.6, force.recalc = TRUE,
                       save.SNN = TRUE, k.param = 25)

# t-SNE plot of calculated centroids, in second plot points are colored
according to subtype
p1 <- TSNEPlot(object = merged, group.by = "organism", do.return =
TRUE, pt.size = 2)
p2 <- TSNEPlot(object = merged, do.return = TRUE, pt.size = 2, group.by
= "phenoinfo")
plot_grid(p1, p2)

 TSNEPlot(object = merged)


# save plots as one pdf file
pdf(file = "yyyymmdd_cca_integration.pdf", onefile = TRUE, width=13,
height=10)

    plot(p1)
    plot(p2)
    plot_grid(p1, p2)

dev.off()
```

4. Clusters

Fetching the information about calculated clusters and producing a cross-table.

```
clusters <- GetClusters(object = merged)
unique(clusters$cluster)

human.clusters <- data.frame(cbind(merged@meta.data$phenoinfo[1:1073],
merged@meta.data$res.0.6[1:1073]))
table(human.clusters)
```

```
# interactive plot
t <- TSNEPlot(merged, pt.size = 2, do.return = T, do.hover =T,
data.hover = "phenoinfo", group.by = "phenoinfo")
htmlwidgets::saveWidget(t, "tsne.html")
```