Analysis of phenotypic and spatial cellular heterogeneity from large scale microscopy data

France Rose

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Analysis of Phenotypic and Spatial Cellular Heterogeneity from Large Scale Microscopy Data

Soutenue par
France Rose
Le 18 Octobre 2019

École doctorale n°
Ecole Doctorale Complexité du vivant ED515

Spécialité
Biologie cellulaire

Composition du jury :

M. Auguste GENOVESIO Ecole Normale Supérieure	*Directeur de thèse*
Mme Priscille BRODIN Institut Pasteur de Lille	*Présidente du jury, Rapportrice*
M. Berend SNIJDER ETH Zürich	*Rapporteur*
Mme Chloé-Agathe AZENCOTT Mines ParisTech	*Examinatrice*
M. Thierry FEUILLET Université Paris 8	*Examinateur*
M. Thierry DORVAL Servier	*Examinateur*

1794
ENS

To my parents and to Don,

Acknowledgements

L'aventure de la thèse ne commence pas le premier jour du contrat doctoral. La mienne a sans doute débuté avec mes parents qui m'ont éveillée au monde, à la démarche et à la curiosité scientifiques, et qui m'ont soutenue lorsque ma motivation a flanché. C'est donc tout naturellement à elleux que mes premiers remerciements s'adressent. Je voudrais remercier le reste de ma famille qui m'aime et me soutient indépendamment de leur degré de connaissance de la recherche et de l'éducation supérieure. De même ces amis de longue date qui m'ont vu évoluer sur le chemin ardu de la thèse : Aurore, Judith, Perrine, Benoît, Thomas, Sophie B., Sophie M., Alice, Virginie et d'autres que j'oublie.

Je voudrais remercier toutes les personnes que j'ai pu croiser à l'IBENS ces 3 années, qui ont fait de cette période une suite de moments conviviaux. Il y en a beaucoup : Lambert, Swann, Amira, Ouardia, Nikita, Toni, Caroline, Maxime, Solène, Alice, Sreetama, Shihav, Clara, Nora... Les ami.e.s qui m'ont précédée et guidée sur le chemin de la thèse (et de la salsa), Joanne et Amhed. Les personnes qui m'ont aidé à soutenir Doc'&co à bout de bras : Joanne, Amhed, Patrick, Charline, Rémi, Emmanuèle - ielles m'ont clairement évité le surmenage ! Je voudrais aussi exprimer un mot particulier pour mon bras droit, Felipe, d'une humeur et d'une gentillesse constantes - et d'une aide inégalée en présentation, illustration, template et conseils de rédaction de thèse ! Mon bras gauche, Mathieu, qui de par sa présence rend le temps moins monotone tous les jours, pestant contre nos ennemis communs : pandas, conda, reviewer 3... Cette thèse n'aurait pas été pareille sans Tiphaine, ma binôme de machine learning, de criblage à haut contenu, de mots fléchés, d'escalade et de vélo. Et récemment Guillaume, qui a sû avec humour et gentillesse se faire une place dans ce planning bien chargé jusqu'à se rendre indispensable.

I want to thank the person that I am most grateful to for these last years. He patiently discussed with me the in and around of academia, publishing, research, professional relationships... His support, love and understanding have made me a better scientist and a better person. Thank you Don.

Je voudrais remercier mon directeur de thèse, Auguste, qui a sû me naviguer pendant cette aventure, me laissant de l'espace pour explorer, m'affirmer, me tromper. And last but not least, the jury members: Berend Snijder, Priscille Brodin for their time and insightful comments on my manuscript; Thierry Feuillet for his communicative interest about spatial statistics and his genuine curiosity for research in Biology; and Thierry Dorval and Chloé-Agathe Azencott for their keen opinions and comments.

Abstract

Robotics and automated fluorescence microscopes have promoted high-content cell-based screenings: fluorescent probes targeting DNA or other major components are used to image hundreds of thousands of cells under many different conditions. Cell-based assays have proven to be efficient at discovering first-in-class therapeutic drugs, i.e. drugs acting on a new target. They allow to detect promising molecules and to profile them, by associating functional annotations to them, like their molecular target or mechanism of action (MOA). I studied heterogeneity of cell responses at different levels and how this phenotypic heterogeneity can be leveraged to better profile drugs. The first level is about studying heterogeneity between patients. We showed that using different patient-derived cell lines increases the chance of predicting the correct molecular target of the tested drug. The second level corresponds to the diversity of cell responses within the same cell line under the same treatment. Appropriate clustering approaches can be used to unravel this complexity and group cells into subpopulations. The proportions of each subpopulation per treatment allow to predict the correct MOA. The third level looks at how the cell subpopulations are spatially organized. I found that neighboring cells influence each others, and display a similar phenotype more frequently than expected at random. These results assessed across a hundred of treatments, show that even genetically identical cells are not all alike and independent, but create spatial heterogeneity via cell lineage and interaction. Using spatial information as well as phenotypic heterogeneity with graph kernel methods improves the MOA classification under some conditions. Alongside, as spatial analysis could be applied on any cell microscopy image, I developed a *Python* analysis package, *pySpacell*, to study spatial randomness from quantitative and qualitative cell markers.

Résumé

La robotique et l'automatisation des microscopes ont ouvert la voie aux cribles cellulaires à haut contenu : des marqueurs fluorescents ciblant l'ADN ou d'autres composants sont utilisés pour imager des centaines de milliers de cellules dans différentes conditions. Il a été montré que les cribles cellulaires sont efficaces pour découvrir des médicaments de nouvelles classes thérapeutiques, cad ceux qui agissent sur une nouvelle cible. Les cribles permettent d'identifier des composés prometteurs et de les caractériser en leur associant des annotations fonctionnelles, comme leur cible moléculaire ou leur mécanisme d'action (MOA). J'ai étudié l'hétérogénéité des réponses cellulaires à différents niveaux et comment cette hétérogénéité phénotypique peut être exploitée pour mieux caractériser les composés. Au premier niveau, j'ai étudié l'hétérogénéité entre patients. Nous avons montré qu'utiliser différentes lignées cellulaires dérivées de patients augmente la probabilité de prédire la cible moléculaire du composé testé. Le second niveau correspond à la diversité des réponses cellulaires de la même lignée cellulaire soumise au même traitement. Des méthodes de clustering appropriées peuvent être utilisées pour clarifier cette complexité et pour grouper les cellules en sous-populations. Les proportions de chaque sous-population par traitement permettent de prédire le bon MOA. Le troisième niveau regarde comment les sous-populations cellulaires sont organisées spatialement. J'ai trouvé que les cellules voisines s'influencent les unes les autres et affichent un phénotype similaire plus fréquemment qu'attendu par chance. Ces résultats obtenus sur une centaine de traitements montrent que des cellules génétiquement identiques ne sont pas identiques et indépendantes mais sont à l'origine d'une hétérogénéité spatiale par le lignage cellulaire et les interactions. En utilisant l'information spatiale ainsi que l'hétérogénéité phénotypique, les méthodes à noyaux de graphes améliorent la classification en MOA sous certaines conditions. Parallèlement, comme l'analyse spatiale peut s'appliquer à n'importe quelle image de microscopie, j'ai développé une librairie d'analyse *Python*, *PySpacell*, pour étudier l'aléatoire spatial de marqueurs quantitatifs et qualitatifs.

Summary

Chapter 1

Introduction

1.1 Big data in Biology

In the last decades, biology practices have turned towards producing and analyzing more and more data. The quantity, speed and quality of data collection have increased in all fields in biology: sequencing and imaging at different scales, from single-cells to whole organisms. For example, the European Bioinformatics Institute (EBI, Hinxton, UK) was storing in 2013, 20 petabytes (1 petabyte = 10^{15} bytes) of data about genes, proteins and small molecules [1], and has now a capacity of storage of 155 petabytes [2]. In the microscopy image realm, acquiring data creates also a significant data stream, especially with modalities like 3D, confocal, light-sheet, or time-lapse. Managing the generated data sets becomes a serious issue, especially as there are currently few commonly accepted data formats and poor interoperability. In addition, there are only a handful of repositories for these data [3–6]. Google has recently launched a beta version for a search engine dedicated to scientific datasets [7], but the quantity of indexed datasets stays minimal for now.

This avalanche of data requires adequate storing, managing and analysis methods, and hence interdisciplinary teams, projects and researchers. Despite the challenges of tackling the amount of data, these data open the possibility to access a wide variety of biological facts: many diverse organisms are now sequenced with high accuracy, variants of already sequenced organisms are added to databases, genetic tests of an unprecedented scale cover the whole genome allowing an unbiased and untargeted search [8], and capturing the variation between single cells or single DNA molecules unravels the diversity inside each organism. Now that the biologist gaze can extend further and further, the goal of cataloguing all the possible states and variations of the living is categorized in the -omic sections: genomics, metabolomics, proteomics, transcriptomics, and even phenomics [9]. Parallel to this goal of uncovering the full space of variations, these amount of data and the integration between different data sources and scales grant the possibility to understand bigger chunks of system-wide biological processes at once.

1.2 The amount of data allows to look for unprecedented complexity and variations in cell phenotypes

In these piles of data, heterogeneity can be observed at different scales, from genes to populations. Furthermore, cell heterogeneity is one of the most important aspect of biological variations, as the cell is the fundamental unit of living organisms [10]. A cell usually has a single genome, produces a common pool of proteins, RNAs and metabolites, and regulates processes via forward pathways and feedback loops. Cell heterogeneity can emerge from changes in the genome, the epigenome, the quantity of each molecule inside the cell, the cellular and extracellular micro-environment, and the morphology [11]. Depending on their type, these differences can create cell subpopulations from different discrete states, e.g. cell cycle phases, survival switching strategies in fluctuating environments [12], or continuous variations, like the level of viral infection [13]. The spatial cues, namely the extracellular matrix, the conditions of culture (2D or 3D), and the cell-cell contacts, create spatial heterogeneity in both tissues and (co-)cultures [14].

The extent of cell variations can be studied in a quantitative manner thanks to the increasing amount of data, and to the automation of data and statistical analyses. Partly, testing different conditions and acquiring a large volume of data for each experiment increases the probability to visualize numerous cell phenotypes.

1.3 High-Content Screening generates a lot of data and is used to discover new pharmaceutical drugs

1.3.1 High-Throughput Screening tests many parallel conditions

High-Throughput Screening (HTS) is an experimental method used especially in fundamental biology research and in drug discovery. It consists in using a controlled set-up to test in parallel up to several thousands conditions (gene knock-outs, drug treatments) in a relatively short time. It became possible with the robotization of multi-well plate handling, parallel liquid handling, and microscopes [15]. It allows to conduct thousands or millions of chemical, genetic or biological experiments in parallel. High-throughput screens are usually used in the first steps of drug design to select a pool of potential compounds, which will be later studied in more details, and from which chemical variants would be synthesized [16].

1.3.2 High-Throughput Screens can be target or phenotypic-based

The first HTS experiments were biochemical, testing small molecules against identified and isolated target proteins, and are called target-based HTS. Yet, another type of HTS was developed, called phenotypic, where cells were seeded in the wells instead of proteins [16]. Major drawbacks of a full biochemical screen are the possible cell toxicity of selected compounds and the possible side-effects when the compound has more than one priority target, causing pleitropic effects. Instead, the phenotypic HTS approach takes the opposite viewpoint, and is able to discriminate at first the cell toxic compounds. It also shows an integrated phenotype on the entire cell, not only on one of its components [17], making possible the direct detection of the main phenotype.

1.3.3 Phenotypic High-Throughput Screening helps discovering new drugs

Image-based phenotypic HTS was recently shown to be a leading technique in the discovery of first-in-class drugs [18]. Drugs are categorized into functional groups describing on which pathway or target they mainly act: these functional groups are called mechanisms of action (MOA). They are, for example, DNA replication or tubulin destabilization. First-in-class drugs have a MOA that is different compared to the already approved drugs. Indeed a meta-study [18] estimated that 37% of first-in-class drugs approved by the US Food and Drug Administration between 1999 and 2008, were discovered through phenotypic-based screening while being only a minority of carried assays. This can be explained by the fact that target-based HTS requires the identification of the cellular target prior to the assay, hence focusing all screening efforts on one pathway.

1.3.4 Phenotypic Screens require downstream drug profiling

When using a phenotypic HTS, the target is not defined, and the screening process is done on the cell phenotype directly. Yet following this screening step, the mechanism of action or the target of the selected compound needs to be investigated: it is the **drug profiling** step. However the complete identification of the biological process is not required to get the drug approved, but facilitates the process greatly. Indeed, in a 2012 study [19], the authors estimated that 7% of approved drugs have no known primary target and 18% of them lack a well defined MOA.

Phenotypic HTS creates a high throughput flow of images, where in most of the cases only simple phenotypes were searched for and detected [20], e.g. the number of cells for cell viability and proliferation assays [21], the nuclear localization of a protein, or the quantity of a fluorescent reporter. From the complexity and the quantity of generated images, the goal was to reduce it to a single number, usually the z-score, for further processing and statistics [22]. However, High-Content Screening (HCS) is about leveraging the quantity of information present in these images [17, 20]. In the different fluorescent channels, the spatial relationships between pixels and their intensities are processed with image and data analysis pipelines [23, 24].

Profiling is an unbiased and sensitive tool to increase the content of the characterized phenotypes via a set of features describing the cell state [25]. The challenge is to find the correct image and data analyses to extract relevant information from the selected features. One widely used drug profiling method is *guilt-by-association* [17]: a compound with unknown MOA is compared to a set of compounds with known MOAs in the same assay (same cell line, same fluorescent markers), i.e. the new compound is compared to each of the described compounds and the new compound is associated to the closest described compound according to the set of chosen features, making the decision that the new compound has the same MOA as the closest described compound. Furthermore, drug profiling can have other goals than identifying a MOA or a molecular target, it can be used to increase the efficiency of future screens, via lead hopping, i.e. selection of chemically different compounds with the same effect as the lead, small molecule library enrichment, and identification of disease-specific phenotypes [25].

1.4 HCS-generated data require specifically designed algorithms

Classical workflows for HCS image analysis follow these basic steps: first segmentation, then feature extraction, and finally biologically-relevant classification [17]. During the usual segmentation, the cells are detected and segmented on the nucleus channel via a peak detection algorithm, then the cell body is segmented via a region growing algorithm from each of the nuclei on the same fluorescence channel or on a dedicated one. Then features derived from shape, moment, intensity, texture and neighborhood measurements are computed for each detected nucleus, cytoplasm, and cell on all the available fluorescence channels [17], and are concatenated inside a numeric vector describing each cellular object. These two first steps can be automated via softwares like CellProfiler [26] or ImageJ/Fiji [27]. The approach for the final step depends

on the biological questions and the fluorescently labeled cellular components. Different methods can be used to find phenotypic similarities: they are divided into supervised and unsupervised methods. The first type, supervised methods or classifiers, learns how to distinguish phenotypes on a manually labeled dataset, then applies what was learned on an unlabeled dataset [28–30]. The second type, unsupervised methods or clusterings, does not need manual annotations and identifies groups of data points only based on the distances or the relations between the corresponding numeric vectors.

For this workflow, the amount of data is an important variable. Indeed, high-throughput methods create a lot of data, and the detection, segmentation and feature extraction algorithms need to compute at a scalable speed. The machine learning methods used to classify the data into pre-existing categories or to reveal their underlying structure also need to be scalable in terms of complexity of computation and time. Additionally, the number of features that can be computed is very large, about several hundreds of features for all regions of interest (like nucleus, cytoplasm, cell). Some features can be redundant or irrelevant to the investigated biological question. The large number of features can cause two main problems for the applied machine learning technique: it can slow down the classifier, but more importantly, it can trick the classifier by making irrelevant features mask relevant data structures (cf Section 2.2.5 about the Curse of Dimensionality). In this case, it is recommended to reduce the number of features via a feature selection or a dimensionality reduction algorithm [17]. Recently, deep learning methods have challenged the classical analysis pipeline. Their performances are now equal or greater compared to other methods [31, 32] (see Section 2.2.4 about supervised methods).

1.5 HCS-generated and other microscopy data are especially relevant to study phenotypic and spatial variations

Once features per cell are extracted from the images, cellular heterogeneity can be studied. Indeed with imaging, cell morphology can be accessed in its full spectrum, via shape measurements and staining fluorescent pattern recognition [33]. Spatial information inside cells and between cells can be easily collected.

The context of HCS is particularly interesting, because it reproduces the same biological experiment multiple times with different conditions, and generates a consequent amount of data. Hence it is an almost ideal setup to study the extent of phenotypic variations created by the different conditions, with a good statistical power.

However, because of the aforementioned complexity of image and data analysis for HCS data, this dimension is usually overlooked, favoring more simple algorithms [20, 34] like the mean vector per treatment instead of dealing with the heterogeneity of single cell vectors.

1.6 Thesis main questions

This cellular heterogeneity can be observed at different scales: between cell lines and inside a same cell line. This heterogeneity is phenotypic and spatial. My focus being drug screening, I looked at how the cellular heterogeneity can improve drug profiling by asking the fol-

lowing questions:

1. Can we use phenotypic heterogeneity between cell lines to predict the molecular target of a drug?
2. Can we improve the drug profiling when adding the spatial arrangement dimension to usual features used in HCS analysis?
3. How to practically analyze the spatial organization of cultured cells from microscopy images?

1.7 Thesis outline

To answer the three main questions, I developed computational methods to analyze, visualize, and compare cellular heterogeneity displayed on microscopy images. I organized this thesis in six main chapters. Namely, in the next chapter (Chapter 2), I described the state of the art of the discipline: from the main biological results implying cell heterogeneity, to the technical methods and challenges this type of image and data analysis is bringing. Then, I presented my results in three chapters:

1. Phenotypic heterogeneity between parallel cell lines can be used for drug profiling. (Chapter 3)
2. Cell spatial arrangement could bring further functional information for drug profiling. (Chapter 4)
3. A Python package for spatial analysis of cell images as a toolbox to answer biological questions. (Chapter 5)

The first (Chapter 3) and the third (Chapter 5) result chapters are presented as papers. The second result chapter (Chapter 4) is detailed without specific format. To finish this thesis, I stated the main conclusions of my work in chapter 6, and discuss these results and future work in chapter 7.

Chapter 2

State of the art

In this chapter, I gather the relevant literature body and previous research results on which my PhD work is grounded. I first review in which context cell-to-cell variability has been observed and what are the current biological paradigms about it. Then I report the methods, both experimental and computational, to detect this cellular phenotypic heterogeneity and their limits. And finally, I examine spatial statistics methods and their applications in cell biology to study spatial heterogeneity.

2.1 Cell-to-cell variability

2.1.1 The causes and consequences of heterogeneity

Definition Phenotypic heterogeneity is discernible in tissues of multicellular organisms, emerging from cell differentiation and specification. This extreme type of heterogeneity has been well established by phenotypical and molecular cues. However, here we will focus on heterogeneity in cultured cells, especially inside a clonal population. Gough and collaborators defined phenotypic heterogeneity as the "variability of one or more phenotypes or traits within a clonal population" [35], and it has been far less studied than tissue heterogeneity. Indeed, most of our biological knowledge comes from population average experiments [36], as for example the level of a protein assessed by Western Blot or the level of a transcript by RNA sequencing.

Possible causes In a clonal cell population, the emerging differences are not for the most part coming from genetic variations. On the contrary, they have a variety of possible causes. They can come from discrepancies in the cell components like the amount of RNAs or the proteins' abundance [36], from different states like cell cycle phases or stress [14, 36], from micro-environmental factors like local cell density or cell-cell contacts [8, 14, 36], and from the discrete nature of molecules and the random nature of molecule-molecule interactions [36]. The relative importance of each of these potential causes is not well assessed. The role played by intrinsic noise is discussed. Indeed the distinction between discrete or continuous states is not clear, as Sacher and collaborators wrote: "Two obvious states are mitosis and apoptosis, but by continuously integrating extrinsic and intrinsic cues, a single cell in a population can have a variety of different states of gene expression and pathway activity." [8] Snijder and Pelkmans argue that there is little intrinsic noise on the opposite, and cell processes are mostly deterministic, with molecular regulatory networks robust to the intrinsic molecular noise [14]. They cite an experiment where cells are grown on small identical fibronectin micropatterns: cells are observed identical in size and shape and the subcellular distribution of intracellular organelles is remarkably constant, "illustrating that the system has little intrinsic noise". Another way to discriminate where the cell-to-cell variability is coming from is to divide it into genetic and non-genetic heterogeneity [8]. Indeed, even in a clonal population, genetic or epigenetic changes happen and are then disseminated through cell division, leading to a cell population composed of genetic mosaics and various signaling pathway activities.

Consequences This cell-to-cell variability causes a diversity of phenotypes.

The internal cell state can be critical for fate determination, i.e. for cell growth, senescence or death [36], for environmental adaptation [12], or for the retinal mosaic for colour vision [37], etc. Not considering cell heterogeneity during analyses has various consequences. Indeed the average may at best only capture the dominant biological mechanism, limiting our understanding of the underlying biological processes [38]. In particular, averaging is not appropriate for single-cell on/off states: for example, individual oocytes present an all-or-nothing maturation in response of a continuously variable concentration of the progesterone hormone, where population-average based models predicted an intermediate level of maturation for an intermediate concentration of progesterone [38, 39]. In the case of the cellular tumor antigen p53, population-average experiments were pointing towards dynamic oscillations of p53 in response to DNA damage, though more recent single-cell experiments uncovered that cells were producing pulses of p53 of fixed amplitude and duration but varying numbers [36]. Hence in variety of cases, population-average conclusions can be misleading, especially in the case of distinct subpopulations. Furthermore, paying attention only to the dominant phenotype may cause the lost of potentially fundamental biological information: subpopulations in minority in the experimental environment can be of major interest in other conditions [8], especially as the culture and experimental conditions may not mimic completely the real case scenario.

2.1.2 How to model heterogeneity: continuous variations or discrete subpopulations

In the previous subsection, we saw the experimental and biological grounds of cell-to-cell variability. This pool of results and facts leads to different ways to model the cellular heterogeneity. The first criterion to decide on, when studying a variant phenotype, is if the observed states are drawn from either a continuous distribution or a discrete one [38].

Different scenarios are possible: the distribution can be homogeneous with or without outliers, display micro-heterogeneity or a macro-heterogeneity with or without outliers. Micro-heterogeneity corresponds to an "apparently continuous random variation in a single phenotype" [35], where macro-heterogeneity corresponds to "variability in one or more cell traits that results in discrete phenotypes and multimodal distribution", which joins with the full discrete model. For the continuous case, the observed phenotypes are displayed in a continuous space and the focus is on variations seen in this distribution [35].

The data exploration tools would be different depending on the model: normality tests or other statistical distribution tests can be used for the continuous model, as well as the percentage of outliers or heterogeneity indices [35]. On the other hand, some techniques are only available under the discrete states assumption like hard clustering methods [17]. Discrete models allow to keep a simple description of the cell heterogeneity when the number of involved features increases: from a long and complex cell representative vector, a single state is assigned to each cell, summarizing different components of the original vector.

It favors associations between different dimensions of the vector: for example, in a hypothetical case, cells with high fluorescence of marker A could also be of bigger size. So even if the underlying state space of phenotypes is thought to be continuous, a discrete approximation could be relevant to study a biological phenomenon.

Cellular heterogeneity modeling is still at its beginning, as most of past approaches consisted in gathering data points by population [20, 34]. The computational techniques are different from the ones used in population-average measurements, and are still being developed. Pioneer studies including cell subpopulations in an HCS data analysis have only been proof-of-principle studies so far. To cite only a few, Loo and collaborators [40] detected cell subpopulations from images by their relative abundance of marked proteins, and illustrated the performance of their method: in a first case study, they showed that they could group treatments in known pathways for HL-60 cells polarization; in a second one, they retrieved correct protein subcellular localization during the cell cycle of H460 lung cancer cell. In a second pioneer study, Fuchs and collaborators [28] created a genome-wide RNAi-induced knockdown assay, identified cell subpopulations, made some predictions for the functions of uncharacterized genes based on phenotypic similarity, and biologically validated one of this prediction for a DNA damage response gene. Past heterogeneity studies in HCS context showed great potential to discover functional associations [25].

2.1.3 The cell-to-cell variability in cancer and diseases models

Cell-to-cell variability is of special importance for cancer and other disease models. Indeed it has been observed that cancer cell responses to drugs can vary widely [41, 42], even with carefully controlled clonal cell populations in laboratories. It is believed that this dynamical and variable response to drugs is a source of resistance to cancer therapies [35], as drug-tolerant cell states can arise through epigenetic changes and non-genetic variability [43–45]. Yet the origins of subpopulations are subject to debate [38], as well as their impact in terms of disease progression and response to therapeutic intervention. Two major models were proposed to explain the origins of this heterogeneity, the cancer stem cell model and the clonal evolution model [46]. In short, the cancer stem cell model hypothesizes that a few cells inheriting from normal stem cells become cancerous and only these have the capacity to contribute to tumor progression and regenerate a tumor. The clonal evolution model hypothesizes that somatic derived cells accumulate mutations and compete among each other, then some acquire self-renewal and tumorigenic properties.

In this context, it is critical to build relevant disease models. It would help to understand cell-to-cell variability and the effects of drugs on it. In return improved knowledge would make clinical models more accurate and the drug screening process more effective.

2.2 Detecting phenotypic heterogeneity in big data

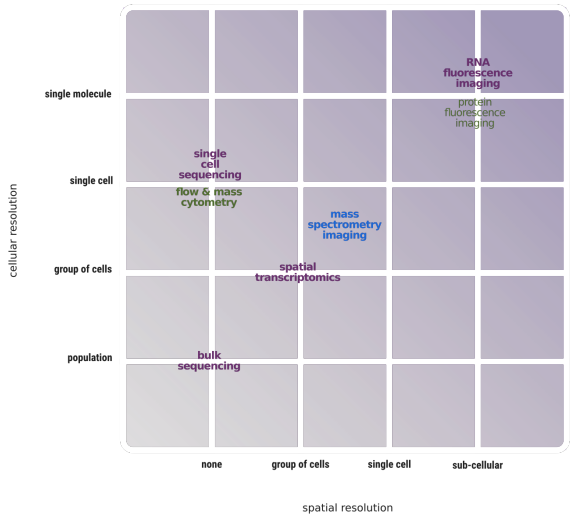


Figure 2.1: **Summary of available single-cell experimental methods.** X-axis represents the spatial resolution in terms of cells, Y-axis represents the degree of single-cell resolution. The color represents the target molecules: RNA/DNA (purple), proteins (green) or proteins & metabolites (blue).

2.2.1 Experiments targeting cell-to-cell variability

Recently there has been such an explosion of single-cell techniques that it is difficult to remain up to date. With the sum of available techniques, single-cell resolution can now be obtained for genetic, epigenetic, spatial, proteomic and lineage data [47]. These methods can be assessed with a few criteria: measured molecule species, single-cell resolution, spatial resolution, throughput in terms of number of cells and in terms of features, and if it is targeted or untargeted [47, 48]. Some methods require a single cell isolation and handling while some others operate *in situ*. To quickly report and summarize the diversity of these methods (cf Figure 2.1), I will divide them into two categories: the first category with methods collecting no spatial data, the second with methods encoding spatial information through visualization *in situ* or retaining spatial origin of the cells via barcodes.

Non-spatial methods

Non-spatial methods require single-cell isolation and handling through micromanipulation, serial dilution, cytometry or microfluidics. I will not go into the details of handling techniques. Lately microfluidics-related methods have improved a lot and been commercialized like "10X genomics" for single-cell RNA sequencing [49]. Single-cell RNA sequencing is now widely used [47]: it is an untargeted method, can measure from 1,000 to 10,000 cells per experiment with around a 10% coverage. Although now, the realm of single cell sequencing techniques has expanded to genomics and epigenomics: indeed the most recent review about single-cell methods published end of January 2019 [47], reports nineteen single-cell sequencing methods covering gene expression (mRNA), genome sequencing, chromatin accessibility, DNA methylation, histone modifications and chromosome conformation. These nineteen methods have varying features and cell throughput, but most of them cover the whole genome or transcriptome with an average cell throughput of 5,000 cells per experiment. Furthermore some very interesting proof-of-concept studies in the two last years demonstrated the feasibility of "multi-omics", measuring two or three data modalities for the same cell, e.g. cellular indexing of transcriptomes and epitopes by sequencing (CITE-seq), is capable of detecting cell surface protein abundance and gene expression in the same cell [47].

On the other hand, cytometry is a widely used technique with a very high throughput, detecting up to thousands of cells per second [50, 51]. It is employed to identify and quantify expression levels of surface and intracellular proteins in single cells. Classical flow cytometry relies on fluorescent markers,

running with ten or more protein markers [50, 52]. However, cytometry also requires the cells to be dissociated in order to be manipulated. One average value is measured per cell and per fluorescent marker. Mass cytometry, also known as cytometry by time-of-flight (CyTOF) [53], relies on antibodies marked with rare metal isotopes which will be detected simultaneously for sensitive identification of proteins. It has a high level of multiplexing, with 40 or more targeted proteins, as there is no problem of fluorescence spectrum overlap or autofluorescence [50, 51, 53].

Spatial methods

Some sequencing techniques have lately incorporated some spatial dimension. Four *in situ* methods are now available: multiplexed error-robust fluorescence in situ hybridization (MERFISH) [54], cyclic single-molecule fluorescence in situ hybridization (osmFISH) [55], spatially resolved transcript amplicon readout mapping (STARmap) [56] and sequential fluorescence *in situ* hybridization (seqFISH) [57]. Their throughputs vary between 100 and 40,000 cells per experiment. It is important to note that these sequencing techniques do not cover the whole genome, and target from 10 to 1,000 RNA species per cell. Another approach from Stahl and coworkers [58] consists in using spatial barcodes during reverse transcription of mRNAs on site, before bulk RNA sequencing. It does not work quite at the single-cell level yet, but has the advantage to be untargeted.

Another modality for spatial single-cell methods is brought by mass spectrometry. At least two techniques, matrix-assisted laser desorption/ionization MALDI and multiplexed ion beam imaging-time of flight MIBI-TOF are able to keep the 2D structure of the sample while extracting ionized molecules. Indeed, the measured spatial area is determined by where the laser or ion beam is aimed at, then the ionized molecules are extracted and determined by the analyzer part of the mass spectrometer. In recent studies [59–61], peptides, lipids and proteins were examined from 2D samples. It is important to note that for peptides and lipids the experimental protocol allowed untargeted detection, whereas for proteins, the protocol included a step of antibody labeling with isotopically pure elemental reporters [60], however still achieving a 40-plex imaging with a 260 nm lateral resolution. This type of methods are also under constant improvement on the detection sensibility and resolution.

More classical forms of fluorescence imaging techniques offer as well single-cell resolution, with gene reporters or antibody-tagged proteins. They operate at cellular high-throughput and are easier to implement than previously described methods, making them suitable for high-throughput screens. Microscopy also has a high-definition spatial representation - usually with subcellular resolution - of cell and organelles shapes, cell types, cell boundaries, cell neighbors [51], giving access to the full spectrum of observable cellular biological phenotypes [33].

2.2.2 Targeted vs untargeted approaches

When comparing targeted to untargeted methods, the first ones usually come with the advantage of detection specificity and precision, but the second ones have a broader coverage and allow to make unbiased discoveries. In the case

of target methods, a priori the only changes that can be detected are the one linked to the used markers. This is partially why there is a constant community effort to make inventory of all the biomarkers corresponding to each cell type and each cancer type [48, 59, 60], to be able to tag and study cellular differences. The limited number of targets makes their selection and the whole experimental planning decisive. For HCS, some studies explored some combinations of fluorescent markers to discriminate the best ones for drug profiling [24, 40, 62, 63]. However, even if only few cellular components can be visualized at the same time, the underlying molecular network is very well connected, and one cellular process is impacted by many others [14, 64]. This is one of the theoretical ground for the Cell Painting assays [65]: this protocol consists of a morphological profiling assay with six multiplexed fluorescent dyes, revealing cellular components and organelles. General components are chosen to get information not only on the targeted elements, but more broadly on general cellular processes. The hypothesis is made that extensive cellular morphology is "rich data source for interrogating biological perturbations, especially at a large scale" [65]. I also based most of my work on this hypothesis, when analyzing high-content screens.

2.2.3 Image preprocessing and feature extraction

Microscopy images of cells require image processing, segmentation and feature extraction. It is especially important, and often mandatory, to automate it when the quantity of data increases. Image analysis is a cornerstone of biological analyses, because it helps ensure accurate, objective and reproducible results [6].

Available softwares and image-analysis libraries

Many tools are available: commercial ones and free open-source ones [6]. Usually commercial tools are sold by microscopy companies along with the imaging instruments. Examples in this category are *MetaMorph* (Molecular Devices), *Imaris* (Oxford instrument), *ZEN* (Zeiss), etc [6]. Free open-source softwares with graphical interface include *BioImageXD* [66], *Icy* [67], *ImageJ/Fiji* [27, 68], *Vaa3D* (3D visualization–assisted analysis) [69], *CellProfiler* [26], *CellCognition* [70], *Ilastik* [71], etc (cf Figure 2.2). They all have their specificity: *Fiji* is favored for electron-microscopy data analysis, *Icy* has some features for behavioral analysis and cell tracking, *Vaa3D* and *BioimageXD* have good 3D features. *ImageJ* has been the longest free open-source available tool, it is the most popular, versatile and widespread image analysis tool. *CellProfiler* was developed for high-throughput data, to be flexible and multi-purpose. *CellCognition* is designed for high-throughput fluorescence microscopy and time-lapse images [70]. Some tools are designed to be collaborative like *ImageJ*, *Fiji*, *Icy*, *CellProfiler*. Indeed one of the reasons of *ImageJ/Fiji* success is the possibility to create plugins and macros, and publish, download and reuse them.

There are also special image analysis libraries that can be used with *Python*, *R* or *Java* programming languages, as *OpenCV* [72], Insight Segmentation and Registration Toolkit (*ITK*) [73], *scikit-image* [74], etc. The programs and toolboxes listed above can be used for image analysis and visualization.

Summary of open-source software discussed in this Review

Software name	Primary function	website
μManager	Image acquisition	http://www.micro-manager.org/
ScanImage	Image acquisition	http://www.scanimage.org/
OMERO	Image database	http://www.openmicroscopy.org/
Bisque	Image database	http://www.bioimage.ucsb.edu/bisque/
OMERO.searcher	Image content search	http://murphylab.web.cmu.edu/software/searcher/
Bio-Formats	Image format conversion	http://www.openmicroscopy.org/
ImageJ	Image analysis	http://rsbweb.nih.gov/ij/
Fiji	Image analysis	http://www.fiji.sc/
BioImageXD	Image analysis	http://www.bioimagexd.net/
Icy	Image analysis	http://icy.bioimageanalysis.org/
CellProfiler	Image analysis	http://www.cellprofiler.org/
Vaa3D	Visualization and image analysis	http://www.vaa3d.org/
FarSight	Visualization	http://www.farsight-toolkit.org/
VTK	Bioimaging library	http://www.vtk.org/
ITK	Bioimaging library	http://www.itk.org/
OpenCV	Bioimaging library	http://opencv.willowgarage.com/wiki/
WND-CHARM	Machine learning	http://code.google.com/p/wnd-charm/
PSLID	Machine learning	http://pslid.org/
Ilastik	Machine learning	http://www.ilastik.org/
CellProfiler Analyst	Machine learning and data analysis	http://www.cellprofiler.org/
PatternUnmixer	Machine learning	http://murphylab.cbi.cmu.edu/software/PatternUnmixer2.0/
CellOrganizer	Machine learning, modeling and visualization	http://cellorganizer.org/
KNIME	Workflow system	http://www.knime.org/

Figure 2.2: **List of available open-source softwares for microscopy acquisition and image analysis.** From [6]

Image analysis steps

Most of the image analysis steps are common between different applications, although different variants of algorithms are possible, and ad hoc parameters selection usually necessary. I will divide it in three steps: illumination correction, segmentation, and feature extraction.

The field-of-view illumination bias is usually coming from a non-uniform light source or optical path [33], it generally adds shading near the borders of the image. There are three main ways to correct for it: take an image with no sample in the foreground and use it to model the uneven illumination, build an illumination model on each image or on an aggregation of many images to see the common pattern. The last one is the most widely used technique.

Then the objects of interest are segmented. Segmentation is the process of locating objects and boundaries, and corresponds practically to assigning a label to every pixel in the image such that pixels with same label are part of the same object. Segmentation methods can be divided in two categories: model-based and machine learning-based [33]. For model-based methods, the decision model is selected by the user, e.g. histogram-based thresholding, Canny edge detection, watershed, as well as the parameters that fit best the data. Usually, the parameters are tweaked manually on a few example images to get the desired output, then applied on the rest of the dataset. In machine learning-based methods, the decision model is selected by the user, e.g. Support Vector Machines (SVM), neural network, random forest, but not the parameters. Instead, a training set is provided with manual annotations corresponding to the objects in the images. Model-based methods are available in almost all softwares and toolboxes listed in the previous section, however machine learning-based methods are, for now, only available in image analysis programming libraries

and few other softwares like Ilastik, but their accessibility is increasing.

Once the objects of interest, like nuclei, cells, and other cell compartments, are segmented, features can be computed on each of them, with the goal of extracting numbers corresponding to the targeted phenotypic characteristics. These characteristics are computed as simple moments from shape measurements and intensity values, and as more complex features from texture and microenvironment such as the number of and distances to neighboring cells. Texture features will extract regularities and periodic changes in pixel intensity values through mathematically defined filters, such as Zernike or Gabor features [26, 33]. All these features would be later on selected and discriminated for the research goal.

Specificity of image analysis in HCS

Image analysis takes on some specificities in the context of HCS. As mentioned in the introduction, most analyses rely on one single descriptor, and images and conditions are ranked according to it [75]. However, cell profiling takes a different approach and computes as many features as possible in order to, in a second phase, select the most robust and biologically meaningful ones [33]. This procedure increases the chances of detecting any change in cell phenotypes, as viewed on the microscopy image.

Because of the high throughput characteristics, the idea is to simplify the segmentation part to reach fast, reliable and accurate algorithms. For this purpose, preliminary assays allow to optimize the seeding concentration, the incubation time and other parameters to a correct range of cell densities, to ease the nucleus detection and to conserve a comfortable number of cells to have reliable features. For example for cell survival assays, if nuclei are too clumped the precision of the measure will be imprecise.

In the case of HCS, experiments are usually carried on 96 or 384-well plates, and the feature values can be impaired by the position of the well. The normalization methods would depend on the design of the plate, i.e. where the controls and the different concentrations of the same compound are located [75]. Also, if several plates are used for the same experiment, a normalization between plates, called batch normalization should be performed. Indeed, if batch effects are not normalized, they can lead to misinterpretation and false conclusions on the effect of a drug for example [33]. Depending on the post processing of the features, they can be normalized through log transformations, z-score normalization or other techniques [33].

The main challenge in HCS is speed and accuracy: with the amount of data, few or no manual post processing and controls can be performed.

Without segmentation approaches

Some methods to extract features without segmentation have been developed for the sake of speed or for images that are hard to segment. Some methods compute classical features on the whole image or random parts of it, like *PhenoRipper* [76] or *WND-Charm* [77]. More recently, deep learning methods have been increasingly available to compute features [78, 79] or to directly classify phenotypes [31]. These deep neural networks are fed whole images, cropped im-

ages, or cropped images centered on previously detected nuclei.

Nonetheless, a good cellular segmentation is at the source of studying heterogeneity at the scale of the cell.

2.2.4 Supervised and unsupervised learning

Once the features extracted, the next step in data analysis is to link feature values to the targeted biological output: this is the realm of learning. Learning methods can be divided in two classes: supervised or unsupervised.

Supervised learning requires a training dataset with manual annotations. These manual annotations are not always available, are expensive to create (experts are required to label biological images or objects), or are inaccurate (several experts can disagree).

On the other hand, unsupervised learning, also known as clustering, does not require a labeled training dataset, and leverages the similarities and dissimilarities between points in the dataset. The limitations of unsupervised learning comprise the evaluation of the quality of the clustering, as well as the choice of the proper feature space where the separation between classes is well represented. Unsupervised learning has the advantage of not being biased by the classes and annotations provided in the training dataset. For cell phenotypes, there are only a handful a striking visual characteristics a human expert can see and label as such, possibly leading to no separation between sophisticated phenotypes or to missed unknown phenotypes [8].

Applying a machine learning method is a key step for cell profiling, as it can discriminate the important features from the unimportant ones, and the reasons of this importance to discriminate the biological output. However, they do not replace a well defined problematic and the choice of a good experimental and computational model to answer it.

These learning methods can be applied to cells or treatments. Most of the time, features are first computed per cell, but are then combined into one vector per treatment [34]. During this process, information is compressed: the output vector corresponds to the "average" cell or to a single-number measure of how the treated cell population differ from the untreated one [29, 80]. However, especially with numerous features, the average vector does not represent a plausible cell but rather a high-dimensionality artifact. As explained in the book "The end of Average" by L. Todd Rose, when taking numerous measurements of different US Air Force pilots and average them measurement by measurement, the average output does not correspond to any existing pilot [81]. It goes the same way for cells, this average found by calculation in a high-dimensional space could not exist in our real life experiment. With this example, the potential for multivariate methods that compare directly populations of cells is clear, creating a more faithful model to the true diversity.

In the next subsections, I will describe some of supervised and unsupervised methods which were previously used for HCS data, or which I applied on my data.

Supervised methods

Support Vector Machine Support Vector Machine (SVM) is a non-

probabilistic binary classifier [82]. This characterization means that a training set with labels corresponding to two classes is needed, and that given a training set the decision boundary will always be the same. Namely, given a training set in a p dimensional space, SVM chooses a hyperplane with p-1 parameters that separates the two classes linearly. Several hyperplanes are suitable, hence the one maximizing the margins, i.e. the distances of the hyperplane to the closest points from each class in the training set, is selected. Once the hyperplane selected, a new point, outside from the training set, can be classified depending on which side of the hyperplane it falls. There are some variants of the SVM. Initially a binary classifier, it can be adapted for more than one class with the one-vs-one and the one-vs-all strategies: a one-vs-one strategy will require $\binom{n}{2}$ classifiers if n is the number of class; a one-vs-all strategy will require n classifiers. Additionally, it might happen that the training set is not separable, then two options are possible: the kernel trick or the soft-margin. The hard margin solution does not authorize any misclassification for points in the training dataset. On the opposite, the soft margin solution authorizes misclassifications but tries to minimize their number. If the data are not linearly separable, even with a few mistakes, the general idea is that the original p-dimensional space can be mapped into a much higher dimensional space where the separation between the two classes is linear. A kernel function is defined such that the computations is done directly from the original space values and no explicit passage to the higher dimensional space is actually needed, hence the name "kernel trick". In the original space, the separation is by way of consequence not linear.

SVMs were broadly used in past studies involving HCS data, and as early as 2006 [28–30, 34, 40, 83–87]. Here, I chose to detail two studies.

The first one is a methodology developed by Loo and others, published in 2007 [29]. In this work, they applied a series of linear SVMs to separate treated and untreated cell populations in the multidimensional space of extracted features. They went further using both the accuracy of each SVM decision and the normal vector to the hyperplane to characterize the transformation of cell populations from the untreated condition to the treated one. They looked at how this accuracy and normal vector change with the increasing dose of each of the 100 compounds they tested. Furthermore, using this normal vector quantifies the phenotypic change caused by the drug. Expressed in the original feature space, this normal vector is a multivariate profile of the phenotypic transformation. Loo and coworkers developed a method based on SVM to extract and compress phenotypic information directly from cell features.

The second one, by Fuchs and coworkers [28], applies SVM on a different level: the hyperplanes are now separating the different cell subpopulations instead of treatments. This approach requires manual annotations, which was not required in the previous approach as the cells were grouped by treatments. In Fuchs' work, about 2,000 cells have been manually classified in phenotypic classes "which included cells showing protrusion/elongation, cells in metaphase, large cells, condensed cells, cells with lamellipodia and cellular debris". Then they summarized each well by the proportion of cells found in each phenotypic class, creating a compressed phenotypic profile. This second method is interesting because it models variations between treatments in terms of

heterogeneity recalling visible subpopulation proportions.

Convolutional Neural Networks Convolutional Neural Networks (CNN) are a subpart of deep learning methods. These networks take directly images as input. They output a label (like MOA) [31], features [78] or a segmented image [88]. Deep learning methods gather neural networks with many layers, where the subsequent layers of processing extract information from the images. The convolutional part refers to the first layers which are designed specifically to extract information from images via convolution of the input images with 2D or 3D filters (depending on the number of fluorescence channels). The great advantage of neural networks is the back-propagation method, which tweaks very efficiently all the parameters distributed in each filter of all the layers in order to decrease the value of a loss function. The loss function increases with the errors made by the CNN measured on the annotated dataset. The back propagation is a deterministic algorithm, in contrast with other optimization algorithm like expectation-maximization or Monte Carlo methods which can be computationally heavy or not converge at all. However, even if the back propagation is itself deterministic, the type of training data and the order in which they are fed to the network may change the learned values of the parameters. On the contrary, in a classical image analysis pipeline, each step has specific parameters that require independent adjustments. Yet, CNN need most of time a large amount of annotated training data, although the needed amount of biological annotated data can be reduced with transfer learning [78, 89, 90].

I will not detail any deep learning method here, as I did not study them specifically during my PhD. They are becoming the uncontested state-of-the-art methods for almost all image applications [91], but they did not seem a good fit to study specifically cell heterogeneity both phenotypic and spatial, as they usually lack interpretability.

Clustering methods

Clustering methods do not need labeled training data, but are sensitive to the structure of the data space, hence to the choice of features. Clustering methods can be exclusive or non exclusive. Exclusive methods, i.e. methods for which a data point only belongs to one cluster, are the most common. Another characteristic of unsupervised methods is the number of levels of clustering: single-level, i.e. one partition of the data points, or hierarchical, i.e. more than one partition of the data points, each one splitting the clusters of the previous one. The methods I present here only deal with numerical data, as the previously described features extracted from cells are numerical and do not include structured ones like strings or graphs. I reviewed classical clustering methods, available in the out-of-the-box *scikit-learn Python* package [74]: k-means, hierarchical clustering, Gaussian Mixture Model (GMM) and spectral clustering, and methods that have been especially designed for high-dimensional data: Louvain's algorithm [92] and Self-Organizing Map [93].

K-means K-means is one of the simplest and most used clustering techniques [82]. It proceeds iteratively starting from a k centers initialization, and repeating the following two steps until convergence: first associating each data

point to its closest center, then redefining each of the centers as the mean vectors of the k groups of data points. K-means procedure tends to find clusters of comparable spatial extent and symmetrical in the different dimensions of data space. It is usually fast and easily scalable with the number of data points and the number of dimensions.

Few attempts on high-dimensional data were made [52, 94, 95]. Ng and collaborators in 2010 used k-means on selected features from the three fluorescence channels separately. This approach allows to reduce the number of input features for each k-mean, and increases the interpretability as an explicit categorization is made on each fluorescent marker first.

However this method is not applied extensively on high-dimensional data because of the underlying hypotheses, namely making groups of the same symmetrical hypervolume, as these hypotheses are probably not valid in high-dimension.

Hierarchical clustering Hierarchical clustering seeks to build a hierarchy of clusters. It proceeds usually through agglomeration of data points then of groups of data points, or splits of the full dataset until singletons. The two main parameters are the metric used to measure the distance between a pair of observations, and the linkage criterion, i.e. how to measure the distance between two sets of points [96].

According to my literature search, hierarchical clustering is widely employed on biological data [30, 94, 97–100]. More precisely, in HCS studies, it happens in a second phase after a processing of the raw image-extracted features: a treatment profile is first computed from the features per cell with another clustering or classification method, then the treatment profiles are compared via hierarchical clustering to find the similarities between them. This way, they can be grouped within gene clusters or MOAs.

DBSCAN Density-based spatial clustering for applications with noise (DBSCAN) is a non-parametric algorithm grouping data points that are closely packed together [101]. It is one of the most common clustering methods and is broadly used in the scientific community. It has many advantages: it is fast, does not require the explicit number of clusters as input, and can detect clusters of different shapes. It only requires two input parameters: ϵ, the maximum distance between two samples in a given cluster, and $MinPoints$, the minimum number of samples in a neighborhood of a point for it to be considered as a core point [102]. It performs better on data containing clusters of similar densities. Unfortunately it has been design to work well on spatial data, i.e. data in a low-dimensional space [103].

Spectral clustering [1]

Spectral clustering has become a popular clustering method in the last years [104]. It operates through a partition of a graph. For numerical data, the preprocessing consists in building a neighboring graph, where each data point is connected to similar points according to a similarity metric and a connecting rule.

The k-nearest neighbors (kNN) connec-

[1] This subsection is mostly based on the very good tutorial by Alexander von Luxburg [104] and the class by Michal Valko from the Master Vision Apprentissage [105].

tion rule consists in putting a link between a data point and each of its k-nearest neighbors. By definition it is asymmetrical, but the practice is to make it symmetrical by taking the union or the intersection of the directed edges. The ϵ distance connection rule links any pair of points whose pairwise distance is smaller than ϵ. These neighboring graphs are respectively called the kNN-graph or the ϵ graph.

Once the similarity graph is drawn between all the data points, computations are done on the Laplacian matrix. The Laplacian matrix is obtained by a simple calculation from the degree matrix and the adjacency matrix. The degree matrix is a diagonal matrix with the degree, i.e. the number of links, of each point. The adjacency matrix is a square matrix with as many lines and rows as there are data points, and where for an index (i,j) the stored value equals 1 if the point i and the point j are linked in the graph, 0 if not.

It can be shown that the number of connected components (CC) in the graph is equal to the multiplicity of the eigenvalue 0 of the Laplacian matrix. Hence to find natural grouping of the data points, several CC are the best case scenario. However, for real data, natural grouping would mean to isolate groups of points intensively connected and loosely connected to other groups of points: each one of these intensively connected groups will translate into one small valued eigenvalue. The clustering in k groups itself is made in the projected space of the eigenvectors corresponding to the k smallest eigenvalues, this projection makes the separation of clusters easy compared to the original space.

This technique is very elegant and has solid mathematical foundations, yet it is not scalable with the number of input points. I found no use of this method for biological high-dimensional data nor for HCS data.

Gaussian Mixture Model The Gaussian Mixture Model (GMM) is a specific type of probabilistic mixture model, where the data set is modeled by k Gaussian components. Each data point is associated with one of the components, assuming that the data point in question is drawn from the model probability distribution. These components are represented, in the case of GMM, by one Gaussian distribution in the original data space. The user chooses a priori the type of model, here Gaussian, and the number of components, k. Each Gaussian model also requires a set of parameters: namely the mean and the covariance matrix. This parameter estimation can be arduous and require a long computational time: the full space of parameters is vast and heuristic methods are applied to make the closest estimation, especially as the number of dimensions of the data space increases. There are two main used methods for this parameter estimation: the expectation-maximization (EM; the one available in scikit-learn) based on maximum likelihood, and the Markov Chain Monte Carlo (MCMC) algorithm based on the maximum a posteriori (MAP). The expectation-maximization algorithm works in two steps: the first one calculates the expectation values for the membership of each data point, then the second one recomputes the distribution parameters based on the newly formed group of data points.

As a mixture model defines subpopulations of points, it is rational to adopt this clustering method when looking for subpopulations of cells [106]. In-

deed, it is widely employed for HCS data [24, 30, 34, 40, 62, 107, 108]. I will detail here two pioneer applications of GMM to HCS data.

Slack and collaborators assumed a cellular heterogeneous response to cancer drugs. For their dataset, cells were treated with drugs and imaged with 3 sets of immunomarkers, one labeling the DNA and the other two some intracellular proteins. Features were obtained from the fluorescence intensity ratios per pixel, averaged in the nucleus and cytoplasm regions. After reducing the number of features to 25, they applied a GMM with k components, representing k cellular subpopulations [24]. They tested several values of k, and compared them in terms of MOA categorization performance, as drugs were associated to MOAs.

In the same year, Yin and coworkers also used GMM to model a heterogeneous cellular response but in an interactive setup [107]. They start with 4 human defined subpopulations: a normal, a long punctuated, a cell cycle arrest and a *rho1* phenotypes. The GMM is initialized on human picked examples of these 4 subpopulations. When a new image with several cells is processed, the newly needed number of clusters is determined through a gap statistic methodology. Then the cells from the new image are associated to existing phenotypes or a new component is added to the GMM. This approach is interesting because it mixes supervised and unsupervised learning, starting from human-defined phenotypes but uses the power of clustering to differentiate novel phenotypes from known ones and clarifying novel phenotypes from each other.

Clustering methods for cytometry data Cytometry data are comparable with image-based HCS data in several ways: it operates at very high throughput (10,000 cells per second or more), reads single-cell values, gets more and more readout values with the improvement of the instruments, and can be used in drug discovery pipelines [50, 109, 110]. The number of possible parallel markers goes routinely up to 7-15 for flow cytometry, and up to 40-100 for mass cytometry [50, 52, 111]. Traditionally, this flow of data has been handled mostly by manual gating, but the increase in throughput and in measured parameters per cell created the need of automated methods crucial [52]. A review by Weber and Robinson provides an overview of the clustering methods used and/or developed by the research community [50]: some are comparable to the ones described above, like k-means (R base package), GMM or other mixture models (SWIFT [112], immunoClust [113], flowClust [114]), spectral clustering (SamSPECTRAL [115]), some use a different approach like flowSOM [52], PhenoGraph [116] or SPADE [117]. I will present in the two next subsections PhenoGraph and flowSOM as they performed well on the benchmarked tests used in this review.

Louvain's algorithm - PhenoGraph implementation The PhenoGraph method [116] is based on the Louvain's algorithm described by Blondel and collaborators, in their article "Fast unfolding of communities in large networks" [92]. The main idea is to construct a nearest neighbor graph (as explained in the spectral clustering paragraph page 27) from the data set, then to partition the graph into sets of highly interconnected points, called

communities. This algorithm falls into the community detection family. It is a heuristic approach based on modularity optimization. In a word, modularity summarizes the concentration of links inside the communities. It is mathematically defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where A_{ij} is the adjacency matrix of the nearest neighbor graph, with values corresponding to weights between a pair of points (i, j), $k_i = \sum_j A_{ij}$ the sum of the weights of the edges attached to point i, c_i is the community the point i belongs to, $m = \frac{1}{2} \sum_{i,j} A_{ij}$ is the normalizing constant, defined as the sum of all weights in the graph, and the δ-function, $\delta(u, v) = 1$ if $u = v$, $\delta(u, v) = 0$ otherwise.

In more details (cf Figure 2.3), the Louvain's algorithm proceeds via an iterative process where two steps are repeated until there are no more changes and a maximum of modularity is reached. Namely, the first step (modularity optimization arrow on Figure 2.3) consists in changing the community membership of points one by one to see if it increases the modularity and keeping the change if it does, the second step (community aggregation arrow on Figure 2.3) merges the points or vertices of a same cluster into one vertex in a reduced graph. On the original research paper from 2008 [92], the Louvain's algorithm has been tested on a mobile phone network of 2.6 million people and a web graph of 118 million nodes and more than one billion links. Hence it is proven to work on large data in terms of number of input data points.

Levine and coworkers were the first one to use it on biological data in 2015 [116], and used *Python* and *Matlab* wrappers of the original *C++* code from Blondel and others. They added in their implementation the graph construction step from a feature matrix. PhenoGraph was originally used to study leukemia cellular heterogeneity via mass cytometry, and isolated a subpopulation that help patient survival prediction.

Self-Organizing Map - flowSOM implementation Self-Organizing Maps (SOMs) are a type of artificial neural network used for unsupervised learning. It learns a mapping between the high-dimensional input space and a low dimensional, usually 2D, discretized space. This low dimensional mapping space is composed of a grid of neurons or nodes, usually a regular hexagonal or rectangular predefined grid. Each node or neuron has a position in the input space, represented by weights. The training consists in changing these weights to fit as best the input data while preserving the topological properties of the mapping space: close neurons on the map will stay close in the input space too. It is trained through competitive learning: when a training example is fed to the network, then the distance to all the weight vectors is computed in the input space, and the neuron chosen, called the Best Matching Unit (BMU), is the closest one. Then the weights of the BMU and close-by neurons are adjusted towards the input training example vector. The magnitude of the change decreases for neurons further in the grid and with training iterations. With this system, the grid contains topological information from the input space, and each training point is influencing several nodes. Once trained, a new vector can be associated to its closest neuron. It is

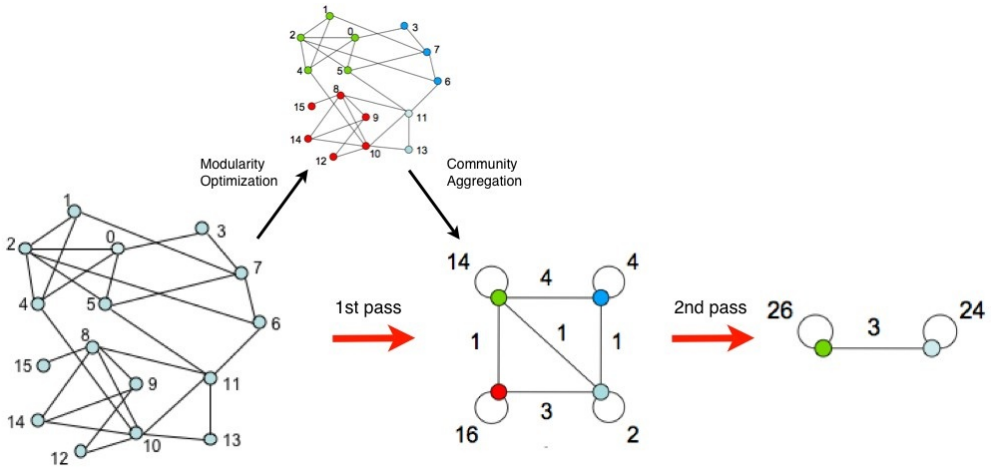


Figure 2.3: **Principle of Louvain's algorithm.** In the input graph (graph on the left), each vertex is its own cluster. In a first modularity optimization step, vertices are grouped together iteratively and the grouping is kept if modularity is increased. This steps ends when changing vertices from one cluster to the other does not increase the modularity anymore, the result is then the top graph. A community aggregation is applied on this graph: putting all vertices from the same cluster into one vertex. The weights are reported on edges and self-loops. These two steps correspond to the first pass. Multiple passes are necessary. The network reaching the best modularity is kept. Figure from [92].

used for data dimensionality reduction and visualization [52].

The flowSOM method for cytometry utilizes the SOM for detection and visualization of rare cell subpopulations. They purposefully build a 2D SOM map with a higher number of clusters than the expected number of cell types in the dataset. They implemented a second clustering step, using Minimal Spanning Tree (MST) to group neurons from the SOM. A MST tries to minimize the sum of the weights of the branches when connecting the nodes, it will group similar nodes together, resulting in a connected acyclic graph.

Testing their method on several real cytometry dataset, they report similar but faster achieved results than Spanning-tree Progression Analysis of Density-normalized Events (SPADE) [117]

Number of clusters' choice A common issue to all clustering methods is to identify the correct number of clusters. Indeed, no ground truth is provided with the training dataset, there is no a priori knowledge on the number of clusters. However, clustering methods can be discriminated between methods that take as input the desired number of clusters and methods that uses other parameters to estimate it. Louvain's algorithm is one of the methods for which the number of clusters is not an explicit input parameter. Indeed, once the similarity graph is built with two parameters, the number k of nearest neighbors and the size of the smallest group, only modularity optimization drives the choice for the number of clusters. On the other hand, most methods require an explicit input number of clusters, for example, k-means, spectral clustering, GMM, SOM. Hierarchical clustering is somehow different as it creates a nested multi-leveled grouping.

To identify an appropriate number of clusters, there are several heuristics. Two of the most used ones are the elbow method and the gap statistics [118]. They both require a measure of the goodness of fit of the clustering: the percentage of variance explained for the

elbow method, or a *Gap* variable built from comparing the change in within-cluster dispersion with the one expected under an appropriate reference null distribution respectively.

The elbow method works under the assumption that at the beginning, adding a new cluster to a few of them will substantially increase the percentage of variance explained, but after adding many clusters, this gain will become marginal, and this is the point that should be chosen under the elbow criterion. It is criticized because the inflection point of the curve is not always easy to determine and the plot of the percentage of variance explained against the increasing number of clusters needs to be further examined.

The gap statistics was defined to be more robust and set a clear decision rule. As it compares the within-cluster dispersion against the one expected under the reference distribution, the chosen number of clusters will be the one maximizing the gap statistics.

Yin and collaborators defined a variant of the gap statistics to reduce bias in the case of unbalanced sized clusters [107]. Spectral clustering has a special built-in trick to decide on the number of clusters: it is recommended to take as many clusters as there are eigenvalues close to 0. Another way to choose the number of clusters consists of having an external way to measure the quality of clustering. For example, for some HCS, mechanisms of action (MOAs) are known for certain compound, and the clustering can be optimized to maximize the MOA classification accuracy.

2.2.5 Machine Learning Challenges

Using machine learning on HCS or other large biological datasets can have some caveats. First of all, the number of data points or the high dimensionality can be inadequate for some algorithms because of the computational complexity of the method. Some methods make assumptions that are not thought to be valid for the considered data, like the shape of the clusters or the balanced number of points per cluster. However, the most important theoretical challenge is the one cause by the high dimensionality, known as curse of dimensionality. It has many consequences for distances, clusterings, and algorithms' behavior. Domingos in his review summarizes it as basic intuitions from 2D or 3D do not hold in high dimensions [119].

Curse of dimensionality

First of all, when the number of dimensions increases, the volume of the space increases exponentially and it is not possible to explore it all. To illustrate this, if d is the dimension of the space and if each dimension (between the min and the max values of the data points) is split into n pieces, the number of hypercubes necessary to cover the space will be n^{d}. For example, with $n = 10$ and $d = 80$, the number of hypercubes 10^{80} is already equal to the number of atoms in the universe. As a consequence, if the space is too large, it is impossible to scan it or to estimate any data density.

Also as d increases, the space becomes sparser and sparser. This sparsity is problematic for any method that requires statistical significance, because to meet the necessary assumptions, the number of data points should also increase exponentially. This is called the

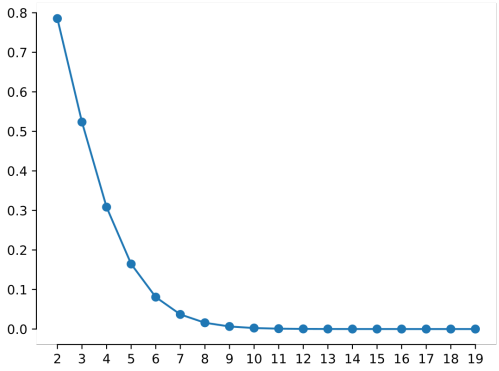


Figure 2.4: **Ratio of the volume of a sphere over the volume of a cube** with increasing dimensionality, from 2 to 19.

"Hugues phenomenon". This sparsity affects all machine learning methods.

Furthermore, for mathematical objects, the periphery is then denser than the central part [119]. For example, a multivariate Gaussian distribution will have more and more weights in its tails as d increases [120], failing to group points in a similar subspace around the mean. To illustrate this, a classical simple ratio of the volume of a sphere a radius r over the volume of a square of radius r equals to

$$V_{sphere}(r) = \frac{\pi^{d/2} r^d}{\Gamma(d/2+1)}$$

$$V_{cube}(r) = (2r)^d$$

$$\frac{V_{sphere}}{V_{cube}} = \frac{\pi^{d/2}}{\Gamma(d/2+1)2^d}$$

The ratio is plotted in the figure 2.4, and from dimension 9, the percentage of volume of the cube contained in the sphere is less than 1 percent, and the majority of the volume fills the corners. This small calculation also shows that some of the consequences of high dimensionality start at quite a low dimension.

A surprising effect of high dimensionality is its impact on distances. There is a general concentration of distances whichever distance metric is used [121]. Every point is at about the same distance as all other points, and the variance of the pairwise distances distribution becomes small compared to the mean. This concentration of distances has a large impact on clustering or other data organization methods: indeed these approaches rely on detecting areas where objects are close according to a distance metric, to deduce their degree of similarity, but in high dimension, data points appear to be sparse and have the same degree of dissimilarity, which prevents common data handling algorithms from being efficient [119].

In more details, not all distance metrics are subjected to this effect to the same degree. L1 distance metric (Manhattan distance) is better suited for high dimensional applications, followed by the Euclidean metric (L2) [122] and by L_k metrics with a higher k. Aggarwal and coworkers suggested fractional norms, i.e. $k < 1$, which are less sensitive to high dimensionality, as they show better contrast between farthest and closest points, however they violate triangular inequality, making them irrelevant for certain applications. This distance concentration phenomenon might be due to the presence of irrelevant "noise" dimensions [123].

Another consequence is the emergence of hubs, points having many neighboring points. It is thought that as points are lying approximately at the same distance of the dataset mean, points lying a little closer to the dataset mean are expected to appear by chance in a non-negligible amount of k-nearest neighbors' list [121]. The distribution of k-nearest neighbors becomes skewed to the right due to these hubs (cf Figure 2.5) [123].

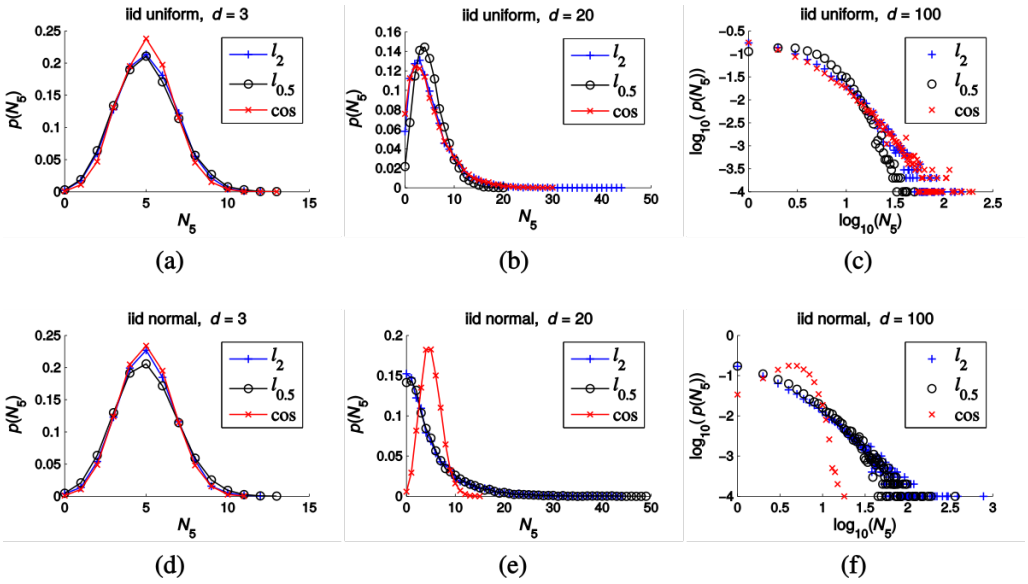


Figure 2.5: **Empirical distribution of the 5-nearest neighbors** N_5 for Euclidean (l_2), $l_{0.5}$, and cosine distances on (a–c) i.i.d. uniform, and (d–f) i.i.d. normal random data sets with n=10000 points and dimensionality (a, d) d=3, (b, e) d=20, and (c, f) d=100 (log-log plot). Figure from [121].

Overcoming the high-dimensionality

The curse of dimensionality has many consequences, happens even for a small number of dimensions (starting from 10), and makes many of the most used methods fail. The above studies usually consider all dimensions as relevant or generate points from uniform or other random distributions, yet real life datasets do not follow all these trends. In most of the cases, dimensions are correlated, and as a consequence, the distances between points display a higher contrast [124]. Also for other high dimensional observations, the effect on real data is less important than in random data.

This usually high correlation found between dimensions can be used to our advantage to decrease the dimensionality without losing information, as the data live in a smaller space [121]. In order to decrease the dimensionality, several methods are available: feature selection methods where a subset of the original dimensions are conserved, and projection methods where a new space with a lower dimensionality is searched for. The optimized criterion for projection methods can be pairwise distances, as for Multi-Dimensional Scaling (MDS), the percentage of the explained variance, as for Principal Component Analysis (PCA), or the local similarities between nearest neighbors, as for manifold learning.

Feature selection methods Features can be selected based on their relevance or their redundancy [125, 126]. Unfortunately, the feature selection problem is ill-posed in an unsupervised context, but these methods are very popular in supervised learning. Indeed in supervised learning, the accuracy of classification or regression is set as an objective criterion for optimization [127–129]. Developing automated feature selection approaches for unlabeled data raises two main issues: first the choice of the number of clusters in relation to the selected feature subset, secondly the need to normalize the selection criterion bias with respect to the dimension [125]. Indeed the goal is to uncover "natural group-

ing" in the data, but it can lead to different clusters for different feature subsets, and different numbers of clusters too. I will focus in this subsection on feature selection for unsupervised learning methods.

Three feature selection frameworks can be distinguished: filters, wrappers and embedded methods [126]. Filters consist in a preprocessing step independent from the downstream of the analysis pipeline. Accordingly, it can only rely on general characteristics of the training data, and make use of the variance, the entropy of univariate dimensions or the correlation between dimensions [130–132]. Wrappers perform the feature selection along with the clustering algorithm in a feedback loop. Indeed the feature selection algorithm uses the learning method to assess the quality of the tested feature subset. They have better performance than filters [126], but can be slower [127] and less generalizable. Embedded methods use feature selection in the process of training, at a lower cost than wrappers, but are not used for unsupervised learning [126]. For the wrappers and the embedded methods, feature subset selection can be done through a complete, a sequential or a random search [128, 132].

Some feature selection methods, mostly filters and wrappers, have been used on HCS data, as reported in the review by Caicedo and collaborators [33]. They cite four main approaches: based on feature correlation, on replicate correlation, on hybrid redundancy and relevance optimization, on feature weighting. Methods based on feature correlation are widely used. The incentive is that correlated features are thought to represent the same underlying biological property, hence one of them could be discarded. It belongs to the filter type, hence is very fast and scalable. Yet, the user needs to provide a threshold over which features are discarded, and a method to choose which feature to discard from a pair or from a group. Also, there are some counter intuitive results showing that even very correlated features are not always redundant [133]. Indeed, transforming variables that appear redundant, by adding or rotating them, may result in noise reduction and better class separation (for examples see [133]).

When working on biological data, use of technical and biological replicates can be effective. Indeed, dissimilarity between replicates should be minimal, or at least dimensions contributing to it should be removed or taken with caution. This idea can be implemented in the form of a filter for feature selection [33]. The two next methods, cited in Caicedo et al. [33], have been developed specifically for HCS data.

Ng and coworkers designed a method minimizing the redundancy while maximizing the relevance of the feature subset. In a nutshell, for each feature a Kolmogorov-Smirnov test is computed on the cell population distributions of feature values between one drug treatment and the negative control. If the KS score averaged on all tested drugs is below a threshold, the feature is discarded. In a second step, pairwise correlation scores between features are computed, and iteratively one feature per pair of highly correlated features is removed. The removed feature is the one with the lowest KS score of the two, showing a lower relevance. In this case, Ng and coworkers used separation between biological conditions (treated cells versus untreated ones) as a criterion.

It is the same assumption that drove Loo and collaborators to propose a recursive feature elimination combined with a support-vector machine (SVM) based

on a wrapper scheme [29]. They extract dimension's weights from the SVMs, which separate treated from untreated cells separately for each treatment. Features with low weights are iteratively removed, until the classification accuracy, meaning the separability between two cellular distributions, drops.

Besides very common methods, like removing highly correlated features, other ad-hoc methods are usually context and dataset-dependent, and may not produce good results for other experiments [34]. It is partly explained by the difficulty to choose a feature subset without an objective measure, which is the case for unsupervised learning.

Projection methods The advantage of feature selection methods is to retain some of the original dimensions which keeps the interpretability, however projection methods can retain a larger part of information while reducing subsequently the number of dimensions. Linear and non-linear methods can be distinguished. Linear methods correspond to principal component analysis (PCA) and related methods like factor analysis or its supervised counterpart, the linear discriminant analysis (LDA). They can be applied on single-cell profiles or aggregated sample-level profiles [33]. PCA projects the data onto a space with orthogonal dimensions that explains best the observed variance. LDA works like a PCA except labels are associated to data points and the model attempts to maximize the between versus within class variance. Factor analysis is also closely related to PCA, however there is no hypothesis about relationships among factors, and it assumes the existence of unobserved latent variables and an unobserved stochastic error term [34]. All three methods are used for HCS profiling [33, 134, 135].

Non-linear projection methods try to preserve pairwise distances as much as possible, mostly for data visualization [136, 137]. They hypothesize that the data lie on a smooth manifold in a lower dimensional space than the original one, hence the denomination of manifold learning. Some of these methods are Isomap [138], local linear embedding (LLE) [139], or t-distributed Stochastic Neighbor Embedding (tSNE) [140]. Isomap and LLE are based on the k-nearest neighbors (kNN). Isomap is sensitive to noise and outliers [141]. It is not applicable to large datasets due to computational complexity [141], while LLE computes faster [120, 136]. However, LLE will collapse points at the origin and only few points are put further away, which provides a non so convenient visualization for similarity grouping. tSNE is now the state-of-the-art method to visualize in 2D or 3D complex datasets [140]. Its goal is to preserve local similarities between points, it achieves it by minimizing the Kullback-Leibler divergence between the origin and the output spaces. It works remarkably well for large datasets, thanks to the Barnes-Hut approximation. Resulting dimensions may be difficult to interpret. Also, the low dimensionality is useful for visualization but may not respect the true dimensionality of the input dataset.

Intrinsic dimensionality estimations Most projections methods need as input the number of output dimensions [142]. This is a key parameter. Indeed if it is chosen too small, important information would be lost; and if it is chosen too large, some noise might be kept in the data [136]. For classification problems, it is more robust to base the dimension choice on the classification er-

ror itself, even if it might have a high computational cost. Unfortunately, in the unsupervised case, as there is no objective classification error, the choice of the lower dimension cannot be based on it.

Here I will list some of the methods to approximate the dimensionality of a dataset, but most of them do not agree for a given dataset, providing non-robust estimations of the true dimensionality. We can distinguish local and global approaches [142].

Local-based methods assume a local linear approximation of the hypersurface on which the data lie, i.e. the tangent space. The approximation of the dimension of this hypersurface is the topological dimension [136, 142, 143].

Global-based methods come from projection techniques (PCA) or geometric ones. When computing a PCA, the value of each eigenvalue is a measure of the percentage of variance explained by each new dimension. However it is shown that PCA tends to overestimate the intrinsic dimension [144]. Geometric methods are also called fractal-based methods because they give a non-integer estimate of the intrinsic dimension. They are all loosely based on nearest neighbors. They are known to fail dramatically for high-dimensional data [142].

It seems that the problem of finding the intrinsic dimensionality of a dataset is not fully resolved especially when the original space has many dimensions. Hence heuristic methods like tSNE are very useful to visualize and get intuition about a multivariate dataset. Likewise, feature selection may be needed for reasons stated above, but there are no strong guarantees that it would improve the following data analysis.

2.2.6 Validation of drug profiling methods

Once the features have been extracted, selected and transformed by a machine learning method, it is time to assess the quality of the experiment and the processing. For this task, a profile similarity metric and a quality assessment process need to be chosen.

Profile similarity measures

HCS profiles, also called fingerprints, are numerical vectors. Hence, metrics can be distances like Euclidean, Mahalanobis, Manhattan, similarities like Pearson's correlation, Spearman's or Kendall's rank correlations, or cosine similarity, or custom learned metrics [28, 33, 134]. As we discussed earlier, Euclidean, Mahalanobis and Manhattan distances are susceptible to the curse of dimensionality, but it depends on how the fingerprint has been built. Distance learning has been applied by researchers from the Boutros lab [28]: a parametrized model is applied on pairs of related proteins and pairs of random proteins, and parameters are optimized to get a smaller distance for related proteins than for randomly paired ones. Reisen and collaborators tested 16 metrics from different categories like distance metrics, linear and non-linear correlation measures, measures comparing sets of up/down regulated features, or connectivity map-like similarity measures [134]. I will report some of their results in the next subsection.

Profiling quality assessment

There are two major ways to assess the quality of the profiling: replicate reproducibility or comparison with ground

truth.

Replicate reproducibility As discussed quickly in the subsection 2.2.5 about feature selection, biological experiments are usually made in replicates to control for biological and technical variability. It is a great tool to assess the data and processing quality. Indeed, replicates should give the same biological conclusions, hence the level of dissimilarity found between replicates is the basal level of interpretable differences. More clearly, if two samples from different treatments are as different as two samples from different replicates of the same treatment, then the two different treatments can not be discriminated according to the data.

Reisen and collaborators in their benchmark showed that besides Manhattan, Euclidean, connectivity map-Kolmogorov-Smirnov the other tested metrics are comparable for full-length fingerprints [134]. However for PCA compressed fingerprints, Kendall or Spearman correlation, or connectivity map like similarity measures perform better. Also for random forest scaling (RFS) compressed fingerprints, Euclidean, Pearson correlation and cosine similarity perform slightly better than other methods. According to this benchmark, there is no similarity measure that surpasses all others for every type of fingerprints.

Ng and collaborators in their proposed pipeline also used the replicate similarity to select the number of final cellular subpopulations [94].

Comparison with ground truth The second way of assessing profiling quality requires ground truth. This ground truth gives cues on which treatments are more similar than others, it could be the assignation of genes in gene pathways, of targets to drugs, or of drugs within mechanisms of action (MOAs) [28, 33, 34, 40, 106]. The idea is similar to the one for replicate similarity: it is assumed that compounds with similar activity induce similar cellular responses, so corresponding fingerprints are expected to be more similar than randomly chosen compounds [134].

In Reisen and coworkers' benchmark, almost no variations between metrics were found for all full-length, PCA-compressed or RFS-compressed fingerprints. The variance of each metric is quite high across tested compounds. They obtained these results computing the receiver operating characteristic (ROC)-area under the curve (AUC), allowing them to measure the enrichment in similar compounds among the top ranks using fingerprint similarity for ranking. Rohban and collaborators used a similar way to compare profiling methods, using enrichment of compounds having the same MOA or gene pathway in the most similar treatments [106].

Besides the enrichment in similar treatments calculation, binary predictions can be computed, as for example, a nearest-neighbor classification procedure which consists in predicting the MOA of the closest profile [34]. To avoid overfitting, all the treatments from the same drug are left out, it is called the leave-one-drug-out (or more generally leave-one-out LOO) cross-validation (CV) procedure, as it is done subsequently for each treatment. The MOA prediction results can be summarized in a confusion matrix or in a global accuracy score. The confusion matrix is a square matrix (cf Figure 2.6) whose size is the number of classification groups, here MOAs. Its lines correspond to the

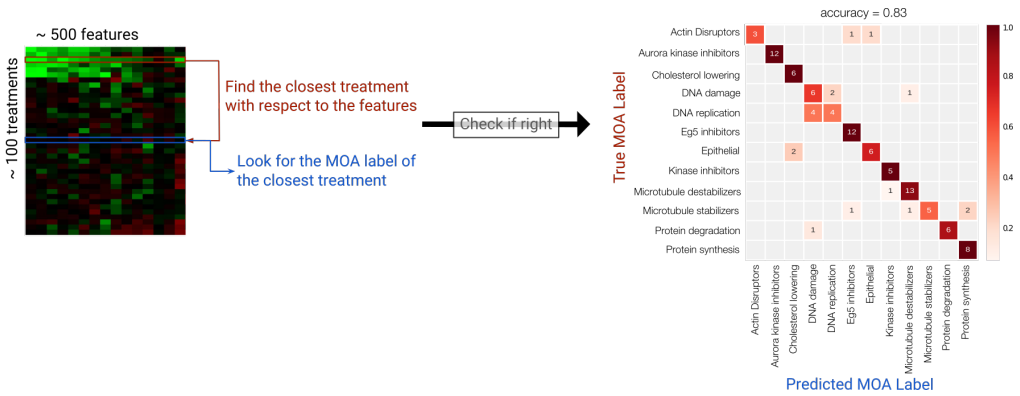


Figure 2.6: **Computation of the confusion matrix from the matrix of feature values per treatment.** For each treatment (red), the closest treatment (excluding the other concentrations of the same drug) with respect to the feature values is found (blue). The true mechanism of action of the treatment (red) is compared to the predicted one (blue), and the corresponding box in the matrix is updated. The global accuracy equals the sum of the diagonal divided by the total sum of the matrix.

true labels, its columns to the predicted labels. If a treatment falls into the diagonal, it means the treatment is classified in the good MOA, otherwise the prediction is wrong. From the confusion matrix, an average accuracy can be computed. As all the MOAs do not have the same number of treatments, there are at least two non equivalent ways to compute the average: on one hand, sum all the treatments falling in the diagonal over the total number of treatments [34]; on the other hand, intermediate averages can be computed per line, i.e. per MOA, and then averaged together [31].

In this section, I discussed state-of-the-art methods to detect phenotypic heterogeneity in big data. Classical pipelines use image preprocessing, segmentation, feature extraction then feature handling. There are many choices to process the features: feature selection or projection, clustering or classification, then validation of the method. I restricted my overview to feature processing for cellular heterogeneity uncovering, such as subpopulation clustering or classification. The quantity of data increases the computational complexity of the processing and its length, but also improves the statistical power such as treatment relationships prediction. The high information content that can be extracted from HCS images show its great potential, but can also be subjected to pitfalls caused by the curse of dimensionality.

2.3 Spatial cellular heterogeneity

2.3.1 A relatively new field in cell biology

Historically only data acquired by microscopy preserve the spatial structure. However only recently the throughput of both microscopes and data analysis pipelines have been able to output large reliable data. The importance of subcellular spatial content was underlined by a study in 2004 [80], where they found that the profiling of drugs using the full range of available microscopy-extracted features performed better than when only using features corresponding to averaged intensities per cell. By this process, they emulated fluorescence-activated cell sorting (FACS) data, and showed that important information would be lost if this technique were used over microscopy. Spatially resolved image data have been available for a long time, and appropriate techniques developed in image

analysis could be of great use for other single-cell resolved techniques, like spatial sequencing or mass spectrometry (cf Section 2.2.1) [145], that have only recently become available [25].

There is no unified spatial analysis found in the literature for cell biology data. Some examples are found in the literature for analysis at a subcellular, tissue or cellular scale. For the subcellular scale analysis, most are based on spatial statistics in 2D or 3D, modeling fluorescence marks as discrete points [146–148].

For the tissue level analysis, methods based on entropy, like the measure of spatial chaos of a mass spectrometry image [149].

At the cellular level, which is the focus of my work, many approaches have been used. Snijder and collaborators defined a handful of micro-environment features that helped them predict the level of viral infection of single-cell from cultured cells [86]. These features were the local cell density, the fraction of cell edges, the distance to the cell colony edge, and some others. After extracting these features, they used a Bayesian decision network to predict the cell viral infection level.

Another work designed a hidden Markov random field (HMRF), that took into account the states of immediate neighboring cells to assign the state of a given cell [150].

A third paper [60] was interested in the mixing of several cell populations, including immune and cancerous cells, they defined an ad hoc mixing score, which was the number of immune-tumor interactions divided by the number of immune interactions. With these three examples, we see that each method was designed to fit one experimental situation, and no common analysis framework was developed.

2.3.2 Spatial statistics

As stated above, no spatial statistics have been developed or widely used for cell biology. Therefore, this literature review will be based mostly on methods employed in geography, ecology and forestry [151–154]. Many mathematical tools have been developed, and some tools used mostly in one of the domains have equivalents in the other domains. A distinction can be made between descriptive statistics suitable for exploratory or descriptive analysis, and modelization tools. Another one can be made between approaches that target numerical variables or categorical ones.

Spatial autocorrelation

One of the basic tool to describe spatial organization for numerical variables is spatial autocorrelation. Basically, it compares the value of a feature for an object and the value of its neighboring objects, normalized by the distribution of the feature [153]. Moran's I is one of these spatial autocorrelation indices [155] and it is defined as following:

$$I = \frac{N}{W} \frac{\sum_i \sum_j w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2}$$

where N is the number of points; x is the feature of interest; $\bar{x}$ is the mean of x; w_{ij} is a matrix of spatial weights; and W is the sum of all w_{ij}.

The weight matrix W encodes the neighboring information between points. For row i, representing node i, the columns filled with a non-null value are the nodes linked with node i, i.e. its neighbors. To scan different sizes of neighborhood,

Moran's I can be computed with several weight matrices linking immediate neighbors or further ones. The result of this scale scan can be represented on a plot called correlogram [156].

The Moran's I defined above gives one value per image, however it can be decomposed in a series of local indices:

$$I_i = \frac{(N-1)(x_i - \bar{x}) \sum_j w_{ij}(x_j - \bar{x})}{\sum_j (x_j - \bar{x})^2}$$

Other very similar spatial autocorrelation indices exist: Getis-Ord's G and Geary's C [154]. These indices are sensitive to outliers, asymmetric data distribution. Feature values are usually normalized before computation [154].

Ripley's functions

Ripley's K function is a very common method to examine the randomness of a spatial point pattern, i.e. if the location of the points are drawn from a totally random spatial pattern or from another distribution. It makes use of distances between all data points and not only between neighbors. The formula for the K function is [157]:

$$K(r) = \frac{1}{N}|A| \sum_{i \neq j} w_{ij}^{-1} 1_r(d_{ij})$$

with A the area, d_{ij} the distance between point i and point j. $1_r(d_{ij}) = 1$ if $d_{ij} \leq r$, $1_r(d_{ij}) = 0$ otherwise. This function is usually compared to the one expected under the complete random spatial (CSR) model, under which $K_{CSR}(r) = \pi r^2$. If objects are more clumped than random, then $\hat{K} > K_{CSR}$, and if objects are more spaced then $\hat{K} < K_{CSR}$.

This function can be extended to bi or multivariate forms, when the points are from few different types, like tree species. For example for two types, the Ripley's K cross-functions are:

$$K_{ij}(r) = \frac{1}{N_i N_j}|A| \sum_{a=1}^{N_i} \sum_{b=1}^{N_j} w_{ab}^{-1} 1_r(d_{ab})$$

Two major hypotheses can be tested with bivariate cross-functions: independance and random labeling [157]. Under the independence hypothesis, events of type i and events of type j do not interact, hence $K_{ij}(r) = K_{CSR}$. If $K_{ij}(r) > K_{CSR}$ then events of type i and events of type j are attracted to each other, on the contrary if $K_{ij}(r) < K_{CSR}$, then the two processes repelled each other. The random labeling approach considers the locations fixed, and allows to question the process assigning labels to points. Under random labeling, we expect $K_{ij} = K_{ii} = K_{ji} = K_{jj}$. Hence departure from the null hypothesis can be studied using pairwise differences between the K cross-functions, e.g. $K_{ij} - K_{ii}$. This inference can be based on Monte Carlo simulations of random labeling to determine the significance of the K functions difference.

Ripley's functions are tools to test both the process that generates the point locations and the process assigning categorical labels associated with the points.

Marked point process methods

Marked point processes correspond mostly to a denomination found in forestry literature [151, 158]. The points are put at tree locations, and these points bear marks such as trunk diameter or tree type. These marks can be of both numerical and categorical types. The two major tools in this framework are the mark correlation and the pair correlation functions. These functions

are different from indices described in the spatial autocorrelation section, as they take the interpoint distance r as a variable. The pair correlation function $g(r)$ is defined as followed [152]:

$$g(r) = \frac{d}{dr}K(r)/(2\pi r)$$

with K the Ripley's function. Contrary to the Ripley's function, the pair correlation function is not cumulative [158], meaning that it focuses on a small variation dr around a value of r. The pair correlation function allows to evaluate the presence of a clustering or an inhibition process at different scales [152]. As its Ripley's equivalent, this pair correlation function only measures the relative locations of points. Additionally, a mark correlation function, as the Ripley's K cross-functions, allows to take into account categorical labels associated with points [158, 159].

Newman's assortativity

In the network science field, the concept of network homophily was defined as the phenomenon when similar nodes are more likely to be linked with each other than dissimilar ones. This definition is based on node attributes. The study of preferential attachment is coming from social sciences, where it is usually observed that people which seem more alike have greater chance to be found in same social circles [160, 161]. As the saying goes, "birds of a feather flock together". Some measures of homophily have been designed [162], and a node attribute assortativity, derived from the more common degree assortativity [163], was defined by Newman in 2003 [161]. This assortativity is a simple way to get a measure for homophily. The weight matrix, storing the links between nodes, is reduced in a smaller matrix with as many rows as there are node classes. The smaller matrix called "mixing matrix" is counting how many links there are between nodes of each category. The mixing matrix (e_{ij}) satisfies the following sum rules:

$$\sum_{i,j} e_{ij} = 1,\ \sum_j e_{ij} = a_i,\ \sum_i e_{ij} = b_j,$$

where a_i and b_i are the fractions of each end node type. If the weight matrix was symmetrical, then the mixing matrix too and $a_i = b_i$.

From this normalized mixing matrix, Newman defines an assortativity coefficient:

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i} = \frac{Tr(\boldsymbol{e}) - ||\boldsymbol{e}^2||}{1 - ||\boldsymbol{e}^2||}$$

where $\boldsymbol{e}$ is the mixing matrix and $||\boldsymbol{e}^2||$ means the sum of all squared elements of the matrix $\boldsymbol{e}$.

When the network is random, the equality $e_{ij} = a_i b_j$ is true, then $r = 0$. If the network is perfectly assortative, all the links fall in the diagonal, hence $\sum_i e_{ii} = Tr\boldsymbol{e} = 1$ and $r = 1$. If the network is perfectly disassortative, no links fall in the diagonal, hence $\sum_i e_{ii} = Tr\boldsymbol{e} = 0$ and $r = -\frac{\sum i a_i b_i}{1 - \sum i a_i b_i} \in [-1, 0[$.

As this node attribute assortativity works with categorical features, it can complement the Ripley's functions.

Other measures

Numerous other indices, functions and tools have been developed, such as the simplest quadrat counts method, nearest neighbor indices and distances (G-, F-, or J- functions) [148, 158], semivariogram representing the variance between pairs of points against their pairwise distance, or spatial regression [164].

All the above methods are considering one feature at a time, this feature being numerical or categorical. Some multivariate approaches exist, such as Mantel's correlation [165], Moran's eigenvector maps, the spatial PCA (sPCA) [166]. The Mantel's correlation is the simplest method and computes a correlation between two distance matrices, one including the distances in the physical space, one including the distances in the feature space. It gives a general idea whether the two types of distances are similar. However, it does not output any details of which feature is participating to this correlation.

Other methods, described in Dray & Jombart's review [166], discriminate features on several criteria, like the variance explained by a spatial partition or by spatial predictors, or the product of variance with the spatial autocorrelation. This last criterion is the one used by spatial PCA: it ranks features according to the value C, defined as followed:

$$C(x) = Var(X)I(x)$$

with $Var(x)$ the variance of feature x in the dataset and $I(x)$ Moran's index. Features with high C and low C values are examined. Indeed, when $C(x)$ is highly positive, $I(x) > 0$ (positive autocorrelation) and the feature displays high variance, and when $C(x)$ is largely negative, $I(x) < 0$ (negative autocorrelation) and the feature also has a high variance.

As PCA, sPCA finds a linear combination of features, but instead of maximizing only the variance, now both the variance and the spatial autocorrelation are under scrutiny [166]. As it is a variant of the PCA, it still has the issue of projection methods where the interpretability is lost in favor of a compressed data visualization.

Comparison to null model

Most of these methods, spatial autocorrelation indices, Ripley's or marked point process functions, are tested against a null model. In the vast majority of cases, the null model is the complete spatial randomness (CSR) model [154, 157, 167]. This complete randomness model assumes a number of points drawn from a Poisson process with intensity λ, and then coordinates of the points are drawn from two uniform distributions on the area A. A Poisson process with intensity λ is a Poisson distribution with parameter (mean) $\lambda * A$.

If the observed data do not follow the CSR, then in a second time clustering or inhibitory processes can be tested [146, 157]. Clustering processes include the Neyman-Scott spatial point process. It operates with a number of parental events, which each produces offspring points. At the end, only offspring points are kept. Offspring points appear around the parental point, sampled from a Gaussian centered on the parent and with a parameterized standard deviation. For this process, there exists an explicit formula for the function K of Ripley, making it easy to fit parameters in order to simulate a matching Neyman-Scott process [146, 157].

In inhibitory processes, there are hard-core and soft-core ones. Hard-core processes, like the Matern point process, do not allow points to be distant from less than some given minimal distance [157]. Soft-core processes, like the Strauss point process, delete the majority of points at a smaller distance than a given critical one [157].

Most of these more complex inhibitory spatial processes are easy to simulate when the target density is not too

high, however if the density increases in the case of inhibitory processes, points will eventually be majoritarily rejected. Moreover, when increasing the complexity of these null models, the number of parameters to fit augments too. Except in the case of the Neyman-Scott process, for which there is a literal expression for the Ripley's K function, the search for appropriate parameter values will be done by expectation-maximization (EM) or Markov chain Monte Carlo (MCMC), which are long and costful algorithms.

On the other hand, the choice of a proper null model is fundamental for the biological questions that can be answered.

Available tools

Some tools, non-specific to cellular data, are available: geographic softwares, like ArcGIS [168], or toolbox like Pysal *Python* module [169], *R* spatstat package [170]. They are a great source of functions and methods.

A few, less versatile, programming toolboxes and GUIs have a biological purposes. RipleyGUI implements Ripley's functions and several null models to analyze spatial patterns of cells in 3D [146]. CellOrganizer is a platform for generative modeling of subcellular components from microscopy images [171]. Very recently, two programming toolboxes have been released in *R* and *Python*: SpatialDE and trendsceek to analyze specifically spatial transcriptomics data. They implement very different approaches: trendsceek is based on marked point process modeling (mark correlation and related functions) [172], whereas spatialDE fits a Gaussian process regression on the 2D coordinates [173]. For network analysis, the *Python* package NetworkX provide some useful functions but is quite limited for the analysis of planar graphs [163].

As spatial data are getting more widely available in cell biology, with the rise of imaging mass spectrometry and spatial single-cell sequencing, more and more methods and toolboxes dedicated to specific spatial analyses will be released in the coming years.

2.3.3 Cell graphs

Most of the spatial statistics presented in the previous section require a weight matrix, like spatial autocorrelation indices, Ripley's K or Newman's assortativity. This weight matrix is square with as many lines as there are objects or vertices in the graph. At each position in the matrix, there is a 0 if the two objects are not connected, and a non-null number if they are. This number can be 1, then the matrix is binary, or a number that represents the strength of the connection. This matrix can be interpreted as an adjacency matrix. Indeed with the information contained in it a graph can be constructed with nodes and edges, and if the matrix is non binary, weights can be added to the edges. In the case where these objects are cells on an image, the graph can even be visualized on the image space (cf Figure 2.7).

Several sets of rules can be applied to build it: neighbors can be selected by Euclidean distance from their center of mass or their borders. Furthermore, depending on the chosen scale, different weight matrices can be built for the same image. As nor spatial statistics, nor the mathematical graph representation have been previously used for cell images, the graph construction modal-

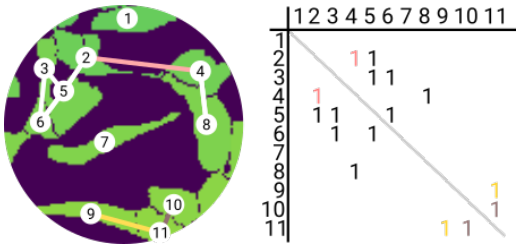


Figure 2.7: **From the cell image to the adjacency matrix of the cell graph.** On the left, a crop of the label image, each shape is a cell. Once detected, cells are assigned a integer label (from 1 to 11 in this example). Binary connections are drawn between cells if their membranes are touching. On the right, a square matrix with the integer labels of cells on the lines and columns. The matrix is filled with 1 if the two corresponding cells are linked on the image, 0 otherwise. Three links on the image and their respective signals in the matrix bear matching colors. When the links between cells are undirected, then the matrix is symmetric.

ities have not been extensively studied in this context. In the geography and ecology literature, they either test several radii with correlogram or mark variogram for example (cf section 2.3.2), or set a scale that would make sense for their analysis, like the borders of a neighborhood (walking distance) or administrative delimitations [174].

Some previous works [175, 176] used this type of cell graph to analyse the difference between cancerous and non-cancerous tissue images. They noticed that the structural and spatial patterns of the cell graph are not random and can be associated with the underlying functional state (cf section 2.3.4) [176].

Usually graphs in biology are used to model signal transduction, regulation, protein-protein interaction, metabolic and chemical pathway networks [177]. The treatment and analysis of these networks are quite different from those suitable for planar graphs like cell graphs. For example, interaction graphs will be studied from the angle of a degree analysis, but the range of node degrees is very limited for planar graphs. Indeed, a cell can only have from 0 to 10 neighboring cells because cells are not point objects and space is limited in 2D. In fact, some proteins are known to have far more interacting bodies than a single digit, this type of graphs are known as scale-free networks [178, 179]. On the other hand, biological questions asked on regulatory networks are very different from the ones deriving from cell graphs: using Bayesian network modelization, the idea is to be able to reconstruct from experimental data the underlying latent regulatory networks and then to be able to predict the outcome of a gene, RNA or protein quantity in relation with time. Cell structural graphs are a measure and simplification of the image and repartition of cells, and beside some parameters like the rules to draw it and the scale, it is very easy to access.

Graph theory has been used in biology, but not to the extent of structural planar graphs, like the cell graphs. Hence the needed methodologies are different as the nature of the graph is different.

2.3.4 Graph comparisons

As the cell graph is a representation of the image, in the context of HCS data, our goal is to compare images, i.e. treatments, by applying a similarity measure and the "guilt-by-association" profiling method (cf Section 1.3.4 in Introduction) [25]. Graph classification can refer to two things: attributing a label to each node, or attributing a label to the whole graph [180]. Sometimes node attributes can be summed or averaged to get the attribute for the graph. Classifying each node as a member of a community is much used in social analysis [181] and a few label propagation algorithms exist, like the one of Zhu and Ghahramani [182]. These are also used in facial recognition, movie recommendation, or epidemy propagation [183].

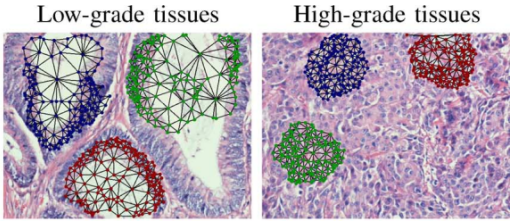


Figure 2.8: **Modelisation of tissue structure by a cell graph.** Different states of tissues, cancerous or not, display different structures. Detail of figure 4 from [186].

However in the general case the problem of comparing graph translates to graph isomorphism. Two graphs G_1 and G_2 are called isomorphic if there is a mapping function f of the vertices of G_1 to the vertices of G_2 such that G_1 and G_2 are identical, i.e. (x, y) is an edge of G_1 if and only if $(f(x), f(y))$ is an edge of G_2 [180]. This problem is known to be Non-deterministic Polynomial (NP)-hard. Even a simpler problem which is subgraph isomorphism, where the goal is to find a subset of vertices of G_1 and a subset of vertices of G_2 is NP-complete [184].

Consequently, the main idea when comparing graphs is to extract features that represent the graph structure and the vertices, and upon which machine learning, like classification or clustering can be applied [181, 185]. The extracted features will make a latent vector representation. This process is called graph representation learning or graph embedding [181]. There are many methods to extract such features. I will present here three main types: hand-defined features, graph kernels and deep neural networks.

Comparing with hand-extracted features

Ozdemir, Gunduz, Yener and collaborators presented interesting results on cancerous cancer tissue detection in a succession of papers [175, 176, 186]: they described the tissue organization as a planar graph where each vertex is a cell connected to its neighboring cells on 2D section biopsies. By an innovative analysis that combines a Voronoi tessellation of space, followed by the extraction of graph related features, like the connected components' size, the number of central points, clustering coefficients, the eigenvalues of the graph Laplacian, the number of endpoints, they trained machine learning algorithms to separate different functional states, like cancerous cells, healthy and unhealthy inflamed cells [175, 176]. These features are mostly based on degree analysis: these features are relevant to distinguish cancerous from non-cancerous tissues because cancerous cells are smaller and proliferate without the usual controls present in a normal tissue, disturbing the organization of the tissue (cf Figure 2.8). In *Python*, numerous features can be computed easily with the package NetworkX [163], although they are primarily based on degree analysis too. Nonetheless, this approach suffered from the same problems as any depending on hand-crafted features: finding appropriate and relevant features for a given classification problem is time-consuming and not always possible. Once the features extracted, any machine learning method can be applied, including feature selection, clustering like k-means, and classification like random forests.

Comparing via graph kernels

Graph kernels are methods defining a kernel that captures properties of interest of graphs for the later application of a kernel classification algorithm such as kernel-Support Vector Machines (kSVM) [180]. These techniques are based on the kernel trick: kernel functions operate in a high-dimensional implicit space without computing the ac-

tual coordinates of the input data in this space, but rather computing inner products between all pairs of data points in the implicit feature space. The kernel trick is used because of its computational advantage. The implicit high-dimensional space is chosen such that the data can be easily clustered or classified in it. Graph kernels can be understood as measuring the similarity between pairs of graphs. They have been introduced in 1999 by Haussler [187]. A good graph kernel must be efficient to compute and applicable to a wide range of graphs [188]. Existent graph kernels compare substructures of graphs that are computable in polynomial time: for example, walks, paths, cyclic patterns, trees.

In biology, graph kernels have been extensively used for protein function prediction and protein-protein interaction prediction. Very handily, graphs can encode both the sequence and the 3D structure of proteins, and cary varying node and edge attributes [189–191]. Ad hoc graph kernels can be defined taking into account the type and length for both edges and nodes when computing the similarity between two graphs. Combining these different kernels into one, parameters of the combination need to be learned for example with a hyperkernel learning [189]. The 3D information is capital to increase the precision of the protein function.

Different graph kernels can be defined, some based on graphlets, which are limited size subgraphs, on subtree patterns, or on walks and paths [192]. Most are available in the graphkernels *R* and *Python* package [193]: label histogram based kernels, random walk based kernels, Weisfehler-Lehman kernels, connected graphlets kernels. Random walk kernels count paths with repetition of nodes of infinite length, but practically, only few nodes constitute the path as the weights assigned on each of the successive node decay. Shortest path kernels compute the shortest distances between all pairs of nodes inside a graph, and have therefore a very high computational complexity. Weisfehler-Lehman kernels are a family of efficient kernels for large graphs with discrete node labels. Connected graphlet kernels count how many occurrences of each type of connected graphlets of a certain size, usually 3 or 4, in a graph. Their principle is detailed in Figure 2.9 [192]. These kernels can or cannot use node labels. They are known to encompass only local properties of graph and ignore the global structure [194].

Comparing via deep neural networks

The revolution brought by Convolutional Neural Networks (CNN) to text and image recognition invited scientists to design new deep neural networks for structured data like graphs. Additionally, graph kernels can be heavy to compute, as they require computing the similarity between each pair of graphs, and computing one similarity can be itself polynomial in the number of nodes. The learning step is separated from the computation of the features - the kernel step - possibly making the features fixed and non optimized for the task. Another problem of graph kernels is that extracted features are not independent [195], as for example a random walk of length l may include random walks of length $l-1$.

Images can be considered as regular grids, and graphs as irregular ones. Hence traditional CNNs would not work on graphs, and extended ones are necessary [194]. Developed CNNs for images are taking advantage of the parsimony

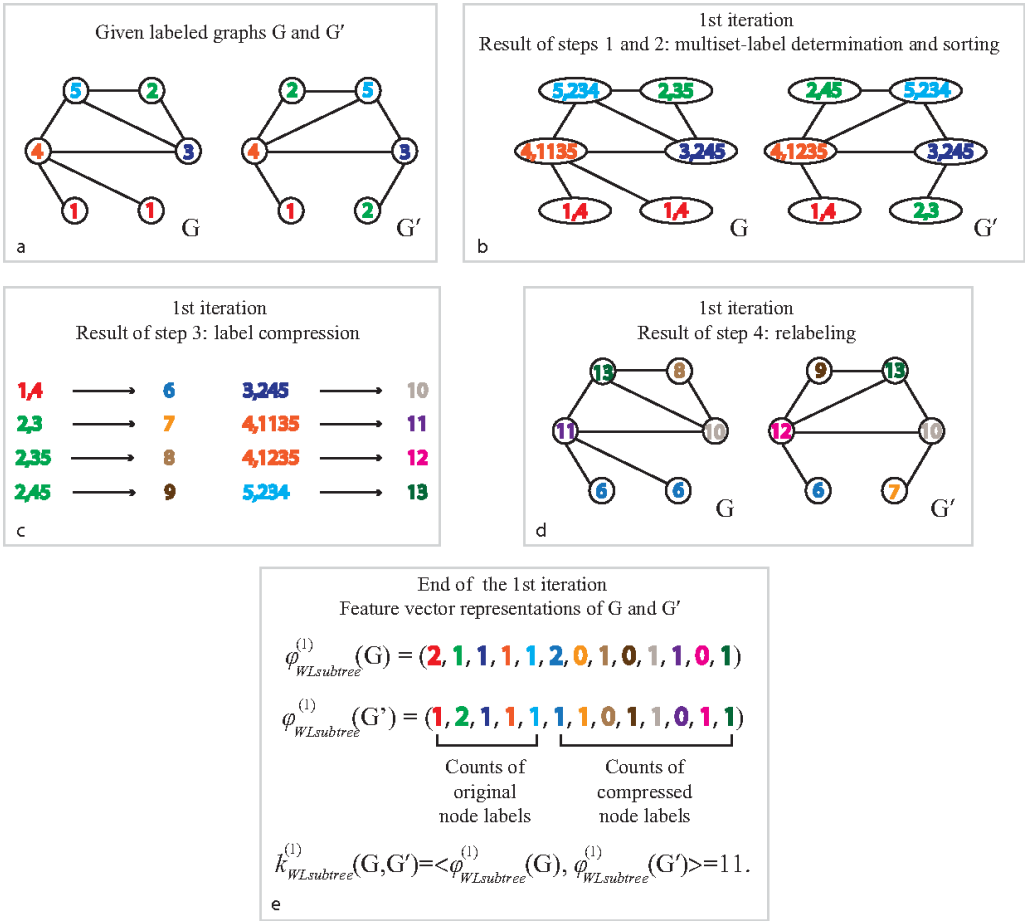


Figure 2.9: **Principle of the Weisfehler-Lehman kernel.** The inputs are two graphs G and G' with vertex labels (a) and a number k of iterations (here for the sake of the explanation $k = 1$). Each iteration consists in listing the labels of the neighboring vertices (b) and concatenating this list into a new label (c) separately for each graph. Relabel the vertices with the new labels (d). After the k iterations, the counts of vertices of each original and new labels are computed and stored in a vector ϕ. The kernel computation between the two graphs is equal to the scalar product of the two vectors ϕ_G and $\phi_{G'}$, outputing a single integer value. Figure from [192].

of parameters, the weights being shared between filters. However, complexity arises with non-regular grids, convolution and pooling are not trivial to define [194]. To overcome this problem, two family of methods can be distinguished: spectral and spatial approaches.

Spectral methods go through the Singular Vector Decomposition (SVD) of the Laplacian matrix of the graph: the eigenvalues of the Laplacian are interpreted as frequencies, and convolutions are carried as pointwise multiplication in the Fourier domain. For example, Li and coworkers [196] used a heat kernel to emulate the convolution, using a sum of products of eigenvalues and eigenvectors from the input graph.

Spatial methods operate directly on the graph structure. In the CNN proposed in Niepert and collaborators [197], graph data are normalized by assuming that neighboring nodes have a fixed ordering, and the same number of neighboring nodes is fed to a neuronal unit. This allows to keep the same number of parameters for a neuronal unit for any input node, preserving the translational invariance of the convolution and hence the relative small overall number of parameters.

However deep learning for graph kernels is still at its early stages, it does not bare natively node or edge labels and are still intractable for large graphs [198].

Chapter 3

Phenotypic heterogeneity can be detected on parallel cell lines for drug profiling.

3.1 Introduction

In this section, I describe the work done in collaboration with two post-doctoral researchers of the team, Elton Rexhepaj and Sreetama Basu, co-authors of this research work. The original idea was to leverage the information stored at the Biophenics screening platform of the Curie Institute [199]. Indeed, as a platform, they perform routinely screens for other researchers in an automated and reproducible way. These screens are usually designed to monitor a single given phenotype, for example they can be used to find drugs which kill cancerous cells efficiently. This is the case for life/death screens where only nuclei are stained in order to count the cells. They can also be used to find drugs stopping the proliferation of cancerous cells. This is the case for an EdU staining protocol. However, with the proper machine learning tool, we expected to be able to combine these past screens in order to predict a drug target or to proceed to pathway enrichment. Theoretically there is no upper limit to the number of cell lines that can be added to the classifier, although we would certainly hit a plateau in terms of prediction performance after a certain number of cell lines combined.

We showed first, that reducing the number of fluorophores does not impair as much as expected the MOA prediction, and secondly, that combining different cell lines treated with the same compounds improves the drug target prediction accuracy with the help of an ensemble classifier. The results are detailed in the following article, that we published in SLAS Technology [200].

3.2 Compound Functional Prediction Using Multiple Unrelated Morphological Profiling Assays

Original Report

SLAS Technology
1–9

DOI: 10.1177/2472630317740831
journals.sagepub.com/home/jla
SAGE

Compound Functional Prediction Using Multiple Unrelated Morphological Profiling Assays

France Rose[1*], Sreetama Basu[1*], Elton Rexhepaj[1,2], Anne Chauchereau[3], Elaine Del Nery[2], and Auguste Genovesio[1]

Abstract

Phenotypic cell–based assays have proven to be efficient at discovering first-in-class therapeutic drugs mainly because they allow for scanning a wide spectrum of possible targets at once. However, despite compelling methodological advances, posterior identification of a compound's mechanism of action (MOA) has remained difficult and highly refractory to automated analyses. Methods such as the cell painting assay and multiplexing fluorescent dyes to reveal broadly relevant cellular components were recently suggested for MOA prediction. We demonstrated that adding fluorescent dyes to a single assay has limited impact on MOA prediction accuracy, as monitoring only the nuclei stain could reach compelling levels of accuracy. This observation suggested that multiplexed measurements are highly correlated and nuclei stain could possibly reflect the general state of the cell. We then hypothesized that combining unrelated and possibly simple cell-based assays could bring a solution that would be biologically and technically more relevant to predict a drug target than using a single assay multiplexing dyes. We show that such a combination of past screen data could rationally be reused in screening facilities to train an ensemble classifier to predict drug targets and prioritize a possibly large list of unknown compound hits at once.

Keywords

target prediction, high-content screening, mechanism of action, ensemble classifier

Introduction

Most often, the current model of drug discovery implies prior identification of a target. This identification allows primary biomolecular screens to focus on a narrowed set of mechanisms of action (MOAs). This approach has shown some success, especially in identifying best-in-class drugs. However, this strategy generates the production of increasingly weak first-in-class drugs, along with ever higher costs. Furthermore, despite widespread adoption of the target-based approach by pharmaceutical companies, an alternative approach, named phenotypic, has brought twice the amount of compounds of a new therapeutic class (i.e., based on a new MOA) to the market in recent years.[1]

Nevertheless, a major challenge of phenotypic approaches remains the posterior identification of the MOA of a lead compound having a desirable effect, for which we have little prior information. Various methods have been developed to dig into the activity of a compound in a cellular system in order to uncover interactions with cell products, including direct biochemical methods, genetic interactions, or computational inference.[2] However, the precise identification of the efficacy target remains a tedious task with little chance of success and is largely refractory to systematic analyses.[3,5]

Inferring the MOA of an unknown compound in a systematic way by phenotypic similarity has been studied in the past.[6,7] More recently, it has been formulated as a classification problem in the phenotypic feature space, using either gene expression or morphometric profiles for a low number of MOAs.[8,10] Drug target associations were predicted this way and experimentally confirmed,[11] suggesting a route to identify the action of new therapeutic agents. Realistically, rather than precisely identifying the efficacy target of each drug, the functional prediction by profiling

[1]Computational Bioimaging and Bioinformatics, Institut de biologie de l'Ecole normale supérieure (IBENS), Ecole normale supérieure, CNRS, INSERM, PSL Research University, Paris, France
[2]Biophenics High-Content Screening Laboratory, Institut Curie, PSL Research University, Paris, France
[3]Prostate Cancer Group, Institut Gustave Roussy, Univ Paris-Sud, Inserm UMR981, Villejuif, France
*These authors contributed equally to this work.

Received June 30, 2017.

Corresponding Author:
Auguste Genovesio, Institut de Biologie de l'ENS (IBENS), 46, Rue d'Ulm, 75005 Paris, France.
Email: auguste.genovesio@ens.fr

could be considered an efficient tool for hit prioritization following a primary high-content screening (HCS). Building a robust method to solve this task on a large scale would bring a solution to one of the main obstacles to phenotypic screening dedicated to drug discovery.

A recent promising approach, the cell painting assay, proposes using six multiplexed fluorescent dyes in a high-throughput image-based assay to create morphological profiles that monitor the general activity of the cell.[12] As for a typical HCS, multi-well-plated cells are treated with compounds, stained, fixed, and imaged with a high-throughput microscope. Automated image analysis is performed to identify individual cells. Then, thousands of morphological features (measures of size, shape, texture, intensity, and neighborhood) on six channels imaging eight cellular compartments are computed to produce a rich profile per cell. Those individual cell profiles can be used for the detection of subtle phenotypes and are efficient to group chemical compounds or genes into functional pathways. Note that adding dyes is limited technically by the number of separated channels one can simultaneously image with a fluorescent microscope, due to the strong overlap of fluorescent protein emission spectra. Therefore, approaches were suggested to identify and iteratively replace the least informative dyes,[13] select the most effective cell line,[14] or image compartments separately in a complex multiple set of experiments.[15] However, the cell painting procedure seems to represents nowadays one of the most optimal ways to simultaneously capture a maximum of information on the cell state in a single high-throughput assay.

In this work, we hypothesize that while a complex cell painting assay on an optimized cell line represents a compelling approach, using a combination of simple image-based assays on several cell lines may be more relevant biologically, more practical, and more accurate for the task of hit prioritization.

Materials and Methods

Patient-Derived Cell Lines

Patient-derived malignant pleural mesothelioma cell lines MPM04/BAR, MPM10/CORO, MPM11/DEL, MPM17/GAG, MPM24/MART, MPM25/MAY, MPM28/MLD, MPM59/LLA, and MPM60/MASS cells (obtained from Didier Jean, INSERM UMR 1162, Paris, France) were cultured in Roswell Park Memorial Institute (RPMI) 1640 medium (Gibco Life Technologies, Waltham, MA), supplemented with 10% (v/v) fetal bovine serum (20% for MPM11/DEL) (Gibco Life Technologies) and 1% (v/v) penicillin/streptomycin (Gibco Life Technologies) in a 5% CO_2 humidified atmosphere at 37 °C. The prostate IGR-Cap1 R100 cell line was grown as described previously.[22] The medium was replaced every 3–4 days.

For compound screening, cells were counted with a T4 Cellometer cell counter (Nexcelom, Lawrence, MA) and then seeded in 384-well plates (ViewPlate-384 Black, PerkinElmer, Waltham, MA) in 40 µL of cell media using the MultiDrop Combi (Thermo Fisher Scientific, Waltham, MA). Cell line densities were empirically determined: 3000 cells per well for IGR-Cap-100; 1750 cells per well for MPM10/CORO, MPM17/GAG, and MPM24/MART; 2000 cells per well for MPM04/BAR, MPM25/MAY, and MPM59/LLA; 2500 cells per well for MPM60/MASS; and 3000 cells per well for MPM11/DEL and MPM28/MLD. The screen was performed at early cell passages for all replicates after the cells had been thawed from liquid nitrogen and passaged three times.

Phenotypic High-Content Screening Assays

Cells were treated with a commercially available collection of 1200 off-patent drugs (the Prestwick library, http://www.prestwickchemical.com). All compounds were diluted in cell media using the MultiChannel ArmTM 384 (MCA 384) (TECAN) for a final concentration of 10 µM in the screen. The concentration of DMSO in each assay well, including all control wells, was 0.5%. Forty-eight hours after compound transfer, mesothelioma cells were next stained with Hoechst 33342 nuclear stain solution (1:500, Life Technologies, Waltham, MA) for 20 min.

Prostate cancer cells were incubated for 3 h at 37 °C with 10 µM 5-ethynyl-2-deoxyuridine (EdU, Invitrogen, Waltham, MA) and then fixed with paraformaldehyde 3% for 15 min (Sigma, St. Louis, MO). Cells were next permeabilized for 5 min with Triton X-100 and labeled with an anti-rat Ki67 cell proliferation marker (1:500, Millipore, Billerica, MA) and DAPI. Click-it reaction was added for 30 min to locate EdU in order to detect cells in the S-phase. The drug screening was performed in two independent replicates for each cell line.

Images were acquired using the IN Cell 2000 automated imaging system (GE Healthcare, Pittsburgh, PA) using a 10× 0.45 NA objective (Nikon, Tokyo, Japan).

For the BBBC021 dataset, details are available in Caie et al.[16]

Image Analysis and Feature Processing

For the mesothelioma screen, each compound was represented by four fields of view (FOVs). CellProfiler[17] was used to initially find nuclei on each FOV from the Hoechst channel. On the identified nucleus mask, a predefined set of 193 features, such as intensity, morphology, and texture, was extracted to describe the drug effect at the single cell level. Features from the different FOVs of the same well were averaged in order to get a single vector describing the compound effect. To normalize for batch and spatial effects, DMSO negative control was used to perform a robust *z*-score normalization and iterative

adaptive median filtering.[23] The two technical replicates were averaged together, defining the final feature representation for each compound.

For the prostate cancer screen, nuclear segmentation from the DAPI channel was used as a mask to extract Ki67 (cell proliferation marker) and EdU (S-phase) marker features. A total of 36 measures were obtained using the IN Cell Analyzer Workstation 3.7 software (GE Healthcare). Then the same normalization as that for the mesothelioma assays has been applied.

For the BBBC021 dataset, Ljosa et al.[10] provided the raw features (453 features) as supplementary data. We followed the same feature processing as described in Ljosa et al.[10] for data normalization and the mean profile method. In more details, we averaged features of cells from the same well (four FOVs per well) in one feature vector. Then we computed the median of the feature vectors corresponding to the same drug compound condition (two to three biological replicates per condition). This procedure gives a unique feature profile for each drug-concentration condition.

Confusion Matrices

MOA annotations for the BBBC021 dataset are available online at https://data.broadinstitute.org/bbbc/BBBC021/ and as supplementary data in Ljosa et al.[10] We used a leave-one-out cross-validation approach; that is, for a left-out compound at a particular concentration, we took its feature profile and determined the nearest-neighbor profile with respect to cosine distance. The predicted MOA for the left-out condition is decided to be the nearest neighbor's MOA label. To avoid overfitting when the nearest-neighboring profile is determined, we excluded other concentrations of the same compound.

Targets of Prestwick Library Compounds

Based on the compound Chemical Abstracts Service (CAS) number, a script was used to automatically query ChemBank and DrugBank simultaneously to retrieve all available data, including all known gene targets for each compound. To mimic the scenario of predicting targets of new compounds, we performed a leave-one-compound-out cross-validation. For this prediction, we could only use a subset of the Prestwick compounds that had at least one common target with the other drugs in the set. From the initial set of 1200 compounds, after removing compounds with no known targets, compounds whose target was unique, or nonclinically validated targets from the original investigator, we obtained 614 compounds targeting 113 genes.

Target Prediction

The prediction task was performed with random forests as weak base classifiers. During training, we constructed a given number of decision trees[24] that we optimized at 30 after testing a range of values between 20 and 40. More precisely for each tree, a random subset of features is selected and the best split of input data points based on this subset is determined to construct the decision tree. To generate predictions, we built an ensemble classifier made of these random forests, with each forest constructed on one assay independently. For this step, we used the MATLAB machine learning toolbox: https://www.mathworks.com/help/stats/classificationensemble-class.html. For a left-out compound, we summed the probabilities to belong to a particular class over all individual random forest classifiers and assigned the maximum class as the predicted class (top 1 prediction). We used this process of soft voting because majority voting would otherwise essentially become a random selection when combining a low number of assays, such as two or three. We also checked if the target was in the top 5 highest soft votes (top 5 prediction).

Supplementary Information

MATLAB code and all datasets needed to reproduce the results are available at https://github.com/biocompibens/MultiplexTarget.

Results

General Fluorescent Markers Seem Redundant for MOA Prediction

We used the BBBC021 dataset[10,16] where cells have been stained for DNA, actin, and tubulin as general markers of the cell phenotype, and treated with 38 drugs at two to three concentrations. An image processing step was performed to detect all individual cells and their nuclei. Following this step, a set of 453 features quantifying intensity, morphology, and texture were computed per cell on all channels using CellProfiler[17] (see Materials and Methods section). Similarly to Ljosa et al.,[10] we used the guilt-by-association approach,[18] which associates an MOA to an unknown compound based on the similarity of their morphometric profiles. Such morphometric profiles were constructed by taking the average value of each feature for all cells of a given treatment, thus producing 453-dimensional vectors when features computed on all channel were considered. In our experiment, we alternatively considered features computed from the nuclei stain only (**Fig. 1A**), features computed from the nuclei and actin stains only (**Fig. 1B**), and eventually features computed from the three available markers (**Fig. 1C**). Note here that the segmentation process for each cell consists of identifying individual nuclei on the DAPI channel first (see examples of DAPI images in **Fig. 2**), and then applying a region-growing algorithm on the actin channel, using nuclei as seeds, to obtain the cell

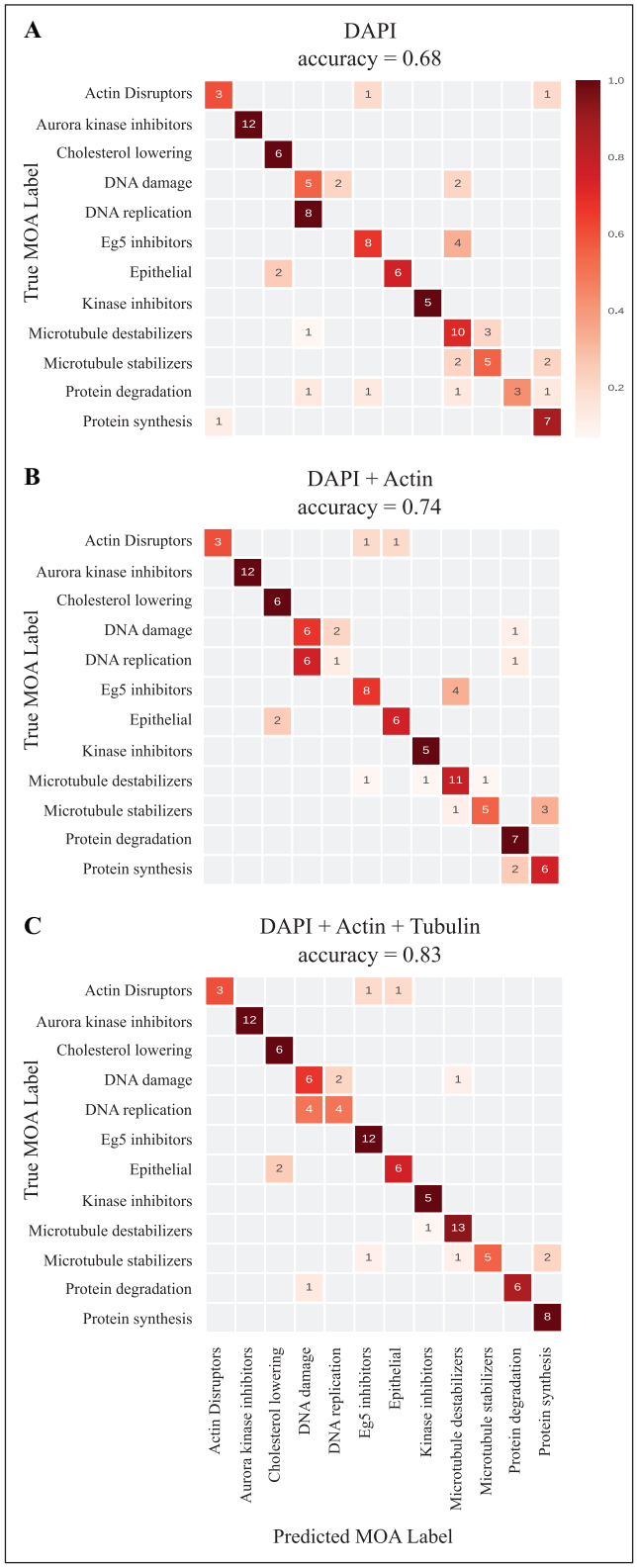


Figure 1. Combining fluorescent markers does not significantly improve MOA prediction accuracy. The above confusion matrices for MOA prediction were obtained using a leave-one-out cross-validation approach. The results are based on features from (**A**) only DAPI, (**B**) DAPI + actin, and (**C**) DAPI + actin + tubulin from the BBBC021 dataset.[16] The predicted MOA for the one-compound concentration condition is the MOA label of its nearest neighbor with respect to cosine distance between feature profiles, as reported in Ljosa et al.[10] Two to three concentrations of 38 drugs form a total of 103 conditions, each related to 1 out of 12 known MOA classes. A value in these confusion matrices gives the number of conditions of a given true label (specified in row) that is predicted as a label (specified in column). The color map is normalized such that each matrix row sums to 1. The accuracy is computed as the ratio between the number of conditions on the diagonal (correctly predicted) and the total number of conditions (103).

boundaries. Therefore, features computed from the tubulin channel depend on the actin channel, which itself depends on the DNA channel. This is the reason why we disregarded profiles that would not contain DNA stain or contained tubulin only. **Figure 1C** reproduces the results obtained in Ljosa et al.[10] using all markers and reaching 83% accuracy for 12 MOAs. Unsuspectingly, **Figure 1A** shows that removing all channels but DNA preserves a high level of accuracy at 68% for 12 MOAs. It indicates that the majority of information is already captured by the nucleus channel and can be retrieved by means of intensity, morphological, and texture features over the cell nuclei. **Figure 2** shows that some of these differences between MOAs are partially visible on the DNA channel. Subsequent addition of actin and tubulin to the DAPI only slightly increases the accuracy for the 12 MOAs (74% and 83%, respectively; **Fig. 1B,C**).

Combining Cell Lines Improves Target Prediction

We reused image sets of screens already performed by the Biophenics platform at the Curie Institute (Paris, France). They included nine patient-derived malignant pleural mesothelioma cell lines and one prostate cancer cell line. We selected those screens mainly because they were all performed with the Prestwick library. This library is a commercially available collection of 1200 Food and Drug Administration (FDA) and European Medicines Agency (EMA) approved drugs that have been extensively studied and for which we could retrieve the known targets through the ChemBank and DrugBank online databases (see Material and Methods). After removing the gene targets for which we had only one corresponding drug or not enough information, we ended with a total of 614 compounds

targeting 113 gene products. We then used a similar image analysis pipeline as in Ljosa et al.,[10] where cells were individually detected, and then features were computed per cell on all available markers (which consisted of DNA only for 9 of the 10 cell lines). Furthermore, mean vector profiles were computed for each sample treatment, that is, per well. We then performed independent trainings of one random forest classifier per cell line and combined the results of all classifiers by soft voting, obtaining a score per target for each compound. We subsequently performed a leave-one-compound-out cross-validation procedure to test the accuracy of our approach in correctly predicting the target of any of the 614 compounds. **Figure 3** shows that the prediction accuracy level is cell line dependent: some cell lines can predict more targets than others. **Figure 3** also demonstrates that prediction accuracy increases with the number of combined cell lines. Altogether, these results show that each cell line brings its share of information, which, when properly combined, can lead to an overall increase of accuracy of the drug target prediction.

Discussion

Using several markers can be useful in a typical HCS assay. Doing so enables protein colocalization to be visualized and specific morphological response to perturbations to be monitored by extracting dedicated features. For instance, it is useful to measure the nuclear translocation of a protein, or a G-protein-coupled receptor (GPCR) internalization. Intuitively, it was reasonable to hypothesize that capturing the general state of the cell would also be better achieved by multiplexing several dyes to monitor principal cellular components, such as endoplasmic reticulum, mitochondria, microtubules, or the actin network, as done in cell painting assays.[12] However, we demonstrated in the Results section that, unexpectedly, increasing the number of fluorescent markers did not substantially improve MOA prediction, as DNA alone seemed to closely characterize directly or indirectly the general state of the cell (**Fig. 2**). Strikingly, the first confusion matrix (**Fig. 1A**) also shows that even MOAs related to cytoskeleton (actin disruptors, microtubule stabilizers, or destabilizers) or other cytoplasmic pathways (protein degradation and cholesterol lowering) could be correctly predicted by DNA staining alone. It was not possible to test yet whether a component other than DNA could be as or more informative on its own because all features computed on other markers depend on DNA staining, which is used to initiate the cell detection algorithm. These results led to the conclusion that DNA shape, texture, and intensity features capture variations similar to those of most other cell components because all these components are interconnected and react together in a perturbation-specific way. In other words, a substantial relationship exists between such global markers, and increasing their amount does not necessarily add significant information for the general task of predicting the MOA or the target of an unknown drug.

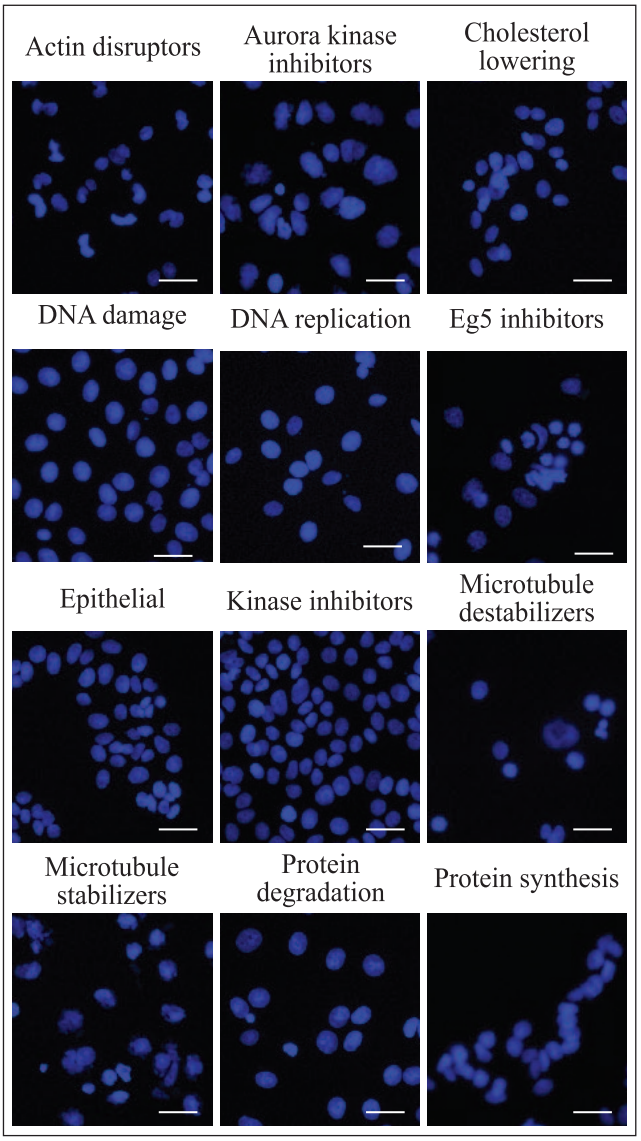


Figure 2. Some MOAs can be distinguished by eye using DNA only. Selected images from each of the 12 MOAs from the BBBC021 dataset.[16] Here, only the DAPI channel is displayed. While it is obviously impossible for the naked eye to precisely sort these images by MOA, differences are clearly discernible in some of the images from the DNA shape and texture. Scale bar, 50 μm.

This observation, combined with the fact that the number of markers that can simultaneously be imaged is technically limited, led to the idea that instead of using multiple dyes on a single cell line, it could be more relevant to use multiple cell lines with few markers. Furthermore, this is precisely the type of screen that can be abundantly found in the screening history of a typical high-throughput platform

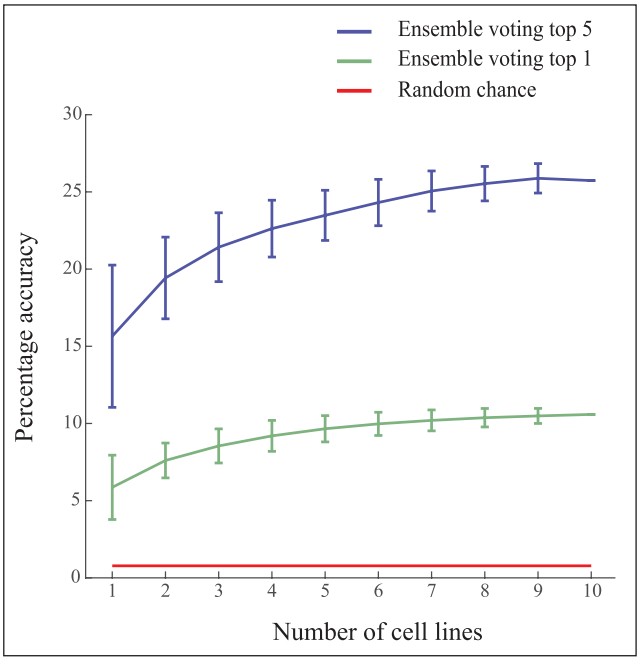


Figure 3. Combining simple assays gradually improves the accuracy of target prediction. Plot showing the accuracy obtained using an increasing number of cell lines for classification. The predicted target for one compound is based on an ensemble classifier trained on 1–10 cell lines. The accuracy is obtained using a leave-one-out cross-validation procedure. Error bars represents standard deviation. The error bar obtained for a single cell line shows that accuracy is cell line dependent; some cell lines, when taken alone, are more predictive than others. For each compound, the ensemble classifier outputs a vector where each value represents the probability of association with a target. The top 1 voting outputs the target corresponding to the mode of this probability vector; the predicted target is then compared with the ground-truth target to determine the accuracy (green line). The top 5 voting outputs the five most probable targets according to the ensemble classifier; the prediction is considered correct when the ground-truth target is present in the top 5 (blue line). These predictions should be compared with the probability to predict the correct target at random, which is 0.09% – 1 over 113, the number of considered targets. Our approach, using 10 random cell lines, achieves 25% accuracy of molecular target prediction out of 113 targets with the top 5 prediction method (blue curve). These results are obtained for 10 cell lines, 614 compounds, and 113 targets.

facility. Accordingly, we gathered data to test this idea. The Results section and **Figure 3** describe how combining a random set of 10 cell-based image assays with few markers can reach better relative accuracy than previously observed using a more complex setup. Indeed, in **Figure 3**, an accuracy 11.3 times better than random is reached (with a value of 0.10 for 113 genes), while Ljosa et al. 10 obtained an accuracy 9.96 times better than random (with a value of 0.83 for 12 MOAs), reproduced in **Figure 1C**.

These results suggested that using multiple cell lines instead of one is more relevant biologically to predict a drug target association. This suggestion matches with the fact that, while highly variable, any given human cell line expresses on average a third of the human genes.[19] As a consequence, even with a known high affinity, a molecule cannot be found to bind to a gene product in cells where this gene is not expressed. Therefore, scanning a larger set of expressed genes through various cell lines would rationally increase the chances of uncovering an existing link between a drug and its target. Using multiple cell lines with simple markers instead of multiple dyes on a single cell line is not only more biologically relevant, but also technically sound, as the number of cell lines that can be added, in opposition to dyes, is virtually unlimited.

Multiplying cell lines would also be practical, as no additional experimental work would be required because past screens can be used as a training set. Typically, most assays performed on an HCS platform only aim to measure a specific phenotype variation, along with the cell count as a measure of toxicity.[20] This cell count is typically based on nucleus staining. Such simple assays are usually cheap, and therefore abundantly available across research institutes and hospital screening facilities. Some previous work described methods where data from high-throughput screening assays stored on PubChem were aggregated to build compound biological fingerprints.[21] They were notably used for biological hit extension by phenotype similarity. However, that approach would not allow us to obtain information on previously uncharacterized compounds that are newly tested. In our approach, we focus on predicting the functions of drugs in order to prioritize a list of hits from HCS assays. We developed a machine learning framework that constructs a classifier from weak learners, each trained on individual assays. As random forest classifiers are independently trained on each assay, our approach can straightforwardly combine assays with a very different nature, number of markers, and quantitative features. In practice, an optimal set of cell lines could be first identified from the past screens of a facility as producing the best ensemble classifier (see the training part in **Fig. 4**); the only requirement is that all those screens use the same library of compounds for which the molecular actions are known. Ideally, the library would include compounds acting on most (if not all) known druggable targets. Prioritizing 200 previously uncharacterized compounds from a new HCS campaign would then consist of thawing cells from the selected cell lines and treating them with the 200 compounds to be characterized (see the application part in **Fig. 4**). Once image analysis and normalization of the features are performed (**Fig. 4B**,**C**,**F**), the classifier trained on past assays would produce a vector of scores for each of the 200 drugs, indicating what gene is most likely to be targeted (**Fig. 4G**,**H**).

The presented results demonstrate that each cell line brings its bit of additional information that incrementally

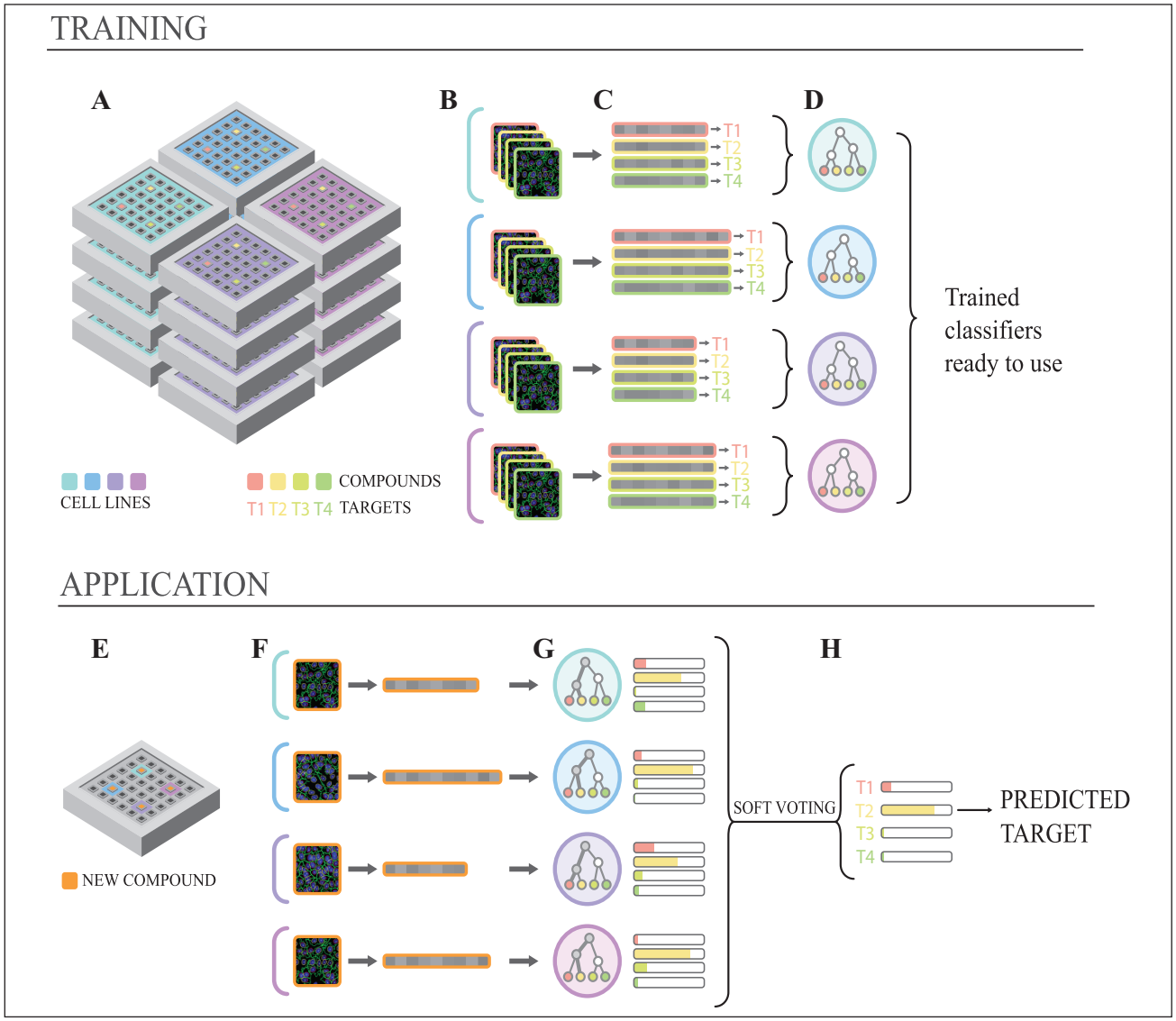


Figure 4. Using past screens for target prediction in practice. (**A**) Past HCS assays performed on different cell lines using the same compound library for which we know the targets. On each cell line, independent (**B**) image processing and (**C**) feature vector extraction are performed. (**D**) Gene targets associated with compounds are used as class labels in combination with feature vectors to train independent classifiers for each cell line. These classifiers learn to associate feature profiles to targets. The training of all classifiers (A–D) is performed only once on the screen history. (**E**) When a set of new compounds (only one is displayed for ease of reading) needs to be characterized in parallel, for instance, to prioritize a set of hits after a primary HCS, (**F**) the same set of cell lines is thawed and used in a smaller secondary screening assay. (**F**) The same image processing and feature extraction pipeline are used. (**G**) Then the already trained classifiers (from D) are used to get one prediction per cell line for each new compound. (**H**) Finally, the predictions from the individual cell line classifiers are combined by soft voting to produce the predicted target(s) for each tested compound.

increases the overall accuracy of drug target prediction. However, there is probably much room for improvement, as 9 out of the 10 cell lines we used were very similar to each other. Those nine cell lines were tumor cells of the same cancer, albeit derived from different patients. Our intuition is that a more diverse set of cell lines corresponding to various tissues would expose a wider range of expression profiles and could possibly largely improve the accuracy of our ensemble classifier. Also, these nine cell lines, in opposition to the remaining prostate cancer cells, were stained with Hoechst only. While we demonstrated that multiplexing global component markers like actin and tubulin may not be very useful, marker labeling for disease model-specific proteins may in turn contribute to characterize

some drug–target interactions. These observations lead to the conclusion that the selection of assays could largely be improved and optimized to maximize accuracy.

An orthogonal track of improvement would consist of considering not only one target corresponding to the best score, but also a set of genes corresponding to the highest scores to identify the pathways targeted by the drug. Indeed, in practice, this approach would be more efficient for two reasons. First, **Figure 3** shows that the chances of finding the target in the five best scores jump to 25% accuracy for 113 targets. Therefore, performing a pathway enrichment analysis may end up being more informative than simply hoping for the efficacy target to get the highest score. Second, it would allow us to identify the pathway targeted by a drug, even if the actual drug's target gene was not part of the training set. The latter is particularly compelling since it would typically be the case for a drug that would bind to a yet uncharacterized target.

In this article, we demonstrated that a common and inexpensive marker, such as Hoechst 33342 or DAPI, labeling DNA, provides unexpectedly rich information on the general state of the cell. We then hypothesized that combining a set of simple past screens could be more efficient to predict the drug target than combining markers in a single assay. We described this approach as not only more relevant biologically and technically, but also applicable in practice to prioritize a list of previously uncharacterized compounds. From this very preliminary proof of concept, we envision two tracks of improvement to be able to claim general applicability of our approach. First, a more diverse combination of cell lines should definitely be selected to expose a larger set of expressed gene products. Second, a set of top-scoring genes could be considered, rather than only the gene corresponding to the best score, in order to robustly recover the targeted pathways. Altogether, we predict that combining simple past assays of an HCS facility could offer a powerful way to prioritize phenotypic hits and alleviate one of the major bottlenecks of phenotypic drug discovery.

Acknowledgments

The authors would like to thank Didier Jean for providing the nine cell lines used for the mesothelioma screens, and Adele Soria, Aurianne Lescure, and Sarah Tessier for their screening technical assistance. We also thank Shantanu Singh, Vebjorn Ljosa, and Anne Carpenter for helpful discussions; Mary Ann Letellier and Don D. Nguyen for valuable comments on the manuscript; and Felipe Delestro from the Bioinformatics Platform of IBENS for the conception and design of the figures.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work has received support under the program "Investissements d'Avenir" launched by the French Government and implemented by ANR with the references ANR–10–LABX–54 MEMOLIFE and ANR–10–IDEX–0001–02 PSL* Research University. Prostate screening was supported by the grant INCAPAIR PROSTATE program n2010-1-PRO-03 from the French National Institute of Cancer.

References

1. Swinney, D. C.; Anthony, J. How Were New Medicines Discovered? *Nat. Rev. Drug Discov*. **2011**, *10*, 507–519.
2. Schenone, M.; Dančík, V.; Wagner, B. K.; et al. Target Identification and Mechanism of Action in Chemical Biology and Drug Discovery. *Nat. Chem. Biol*. **2013**, *9*, 232–240.
3. Liu, X.; Zhu, F.; Ma, X.; et al. The Therapeutic Target Database: An Internet Resource for the Primary Targets of Approved, Clinical Trial and Experimental Drugs. *Expert Opin. Ther. Targets* **2011**, *15*, 903–912.
4. Santos, R.; Ursu, O.; Gaulton, A.; et al. A Comprehensive Map of Molecular Drug Targets. *Nat. Rev. Drug Discov*. **2016**, *16*, 19–34.
5. Keiser, M. J.; Roth, B. L.; Armbruster, B. N.; et al. Relating Protein Pharmacology by Ligand Chemistry. *Nat. Biotechnol*. **2007**, *25*, 197–206.
6. Perlman, Z. E.; Slack, M. D.; Feng, Y.; et al. Multidimensional Drug Profiling by Automated Microscopy. *Science* **2004**, *306*, 1194–1198.
7. Lamb, J.; Crawford, E. D.; Peck, D.; et al. The Connectivity Map: Using Gene-Expression Signatures to Connect Small Molecules, Genes, and Disease. *Science* **2006**, *313*, 1929–1935.
8. Feng, Y.; Mitchison, T. J.; Bender, A.; et al. Multi-Parameter Phenotypic Profiling: Using Cellular Effects to Characterize Small-Molecule Compounds. *Nat. Rev. Drug Discov*. **2009**, *8*, 567–578.
9. Berg, E. L.; Yang, J.; Polokoff, M. A. Building Predictive Models for Mechanism-of-Action Classification from Phenotypic Assay Data Sets. *J. Biomol. Screen*. **2013**, *18*, 1260–1269.
10. Ljosa, V.; Caie, P. D.; Horst, R.; et al. Comparison of Methods for Image-Based Profiling of Cellular Morphological Responses to Small-Molecule Treatment. *J. Biomol. Screen*. **2013**, *18*, 1321–1329.
11. Campillos, M.; Kuhn, M.; Gavin, A.-C.; et al. Drug Target Identification Using Side-Effect Similarity. *Science* **2008**, *321*, 263–266.
12. Bray, M.-A.; Singh, S.; Han, H.; et al. Cell Painting, a High-Content Image Based Assay for Morphological Profiling using Multiplexed Fluorescent Dyes. *Nat. Protoc*. **2016**, *11*, 1757–1774.
13. Loo, L.-H.; Lin, H.-J.; Steininger, R. J.; et al. An Approach for Extensibly Profiling the Molecular States of Cellular Subpopulations. *Nat. Methods* **2009**, *6*, 759–765.

14. Kang, J.; Hsu, C.-H.; Wu, Q.; et al. Improving Drug Discovery with High-Content Phenotypic Screens by Systematic Selection of Reporter Cell Lines. *Nat. Biotechnol*. **2016**, *34*, 70–77.
15. Reisen, F.; Sauty de Chalon, A.; Pfeifer, M.; et al. Linking Phenotypes and Modes of Action through High-Content Screen Fingerprints. *Assay Drug Dev. Technol*. **2015**, *13*, 415–427.
16. Caie, P. D.; Walls, R. E.; Ingleston-Orme, A.; et al. High-Content Phenotypic Profiling of Drug Response Signatures across Distinct Cancer Cells. *Mol. Cancer Ther*. **2010**, *9*, 1913–1926.
17. Carpenter, A. E.; Jones, T. R.; Lamprecht, M. R.; et al. CellProfiler: Image Analysis Software for Identifying and Quantifying Cell Phenotypes. *Genome Biol*. **2006**, *7*, R100.
18. Boutros, M.; Heigwer, F.; Laufer, C. Microscopy-Based High-Content Screening. *Cell* **2015**, *163*, 1314–1325.
19. Ramsköld, D.; Wang, E. T.; Burge, C. B.; et al. An Abundance of Ubiquitously Expressed Genes Revealed by Tissue Transcriptome Sequence Data. *PLoS Comput. Biol*. **2009**, *5*, e1000598.
20. Singh, S.; Carpenter, A. E.; Genovesio, A. Increasing the Content of High-Content Screening: An Overview. *J. Biomol. Screen*. **2014**, *19*, 640–650.
21. Cortes Cabrera, A.; Lucena-Agell, D.; Redondo-Horcajo, M.; et al. Aggregated Compound Biological Signatures Facilitate Phenotypic Drug Discovery and Target Elucidation. *ACS Chem. Biol*. **2016**, *11*, 3024–3034.
22. Al Nakouzi, N.; Cotteret, S.; Commo, F.; et al. Targeting CDC25C, PLK1 and CHEK1 to Overcome Docetaxel Resistance Induced by Loss of LZTS1 in Prostate Cancer. *Oncotarget* **2014**, *5*, 667–678.
23. Bushway, P. J.; Azimi, B.; Heynen-Genel, S. Optimization and Application of Median Filter Corrections to Relieve Diverse Spatial Patterns in Microtiter Plate Data. *J. Biomol. Screen*. **2011**, *16*, 1068–1080.
24. Breiman, L. Random forests. *Mach. Learn*. **2001**, *45*, 5–32.

3.3 Discussion and perspectives

The work reported here is mostly a proof of concept. Nine of the ten cell lines we used have a very close genetic background, as they are all patient-derived malignant pleural mesothelioma cell lines, and only one of them is from a different cancer type, namely prostate cancer. Currently, Maxime Corbe, a research engineer working in collaboration between our team and the Biophenics platform, is building a web interface to easily access and visualize past screens and their outputs, e.g. the z-score, the cell counts, etc. This tool will allow to put in practice and improve our method that uses past screens to predict drug targets. Indeed, this web graphical interface will ease the identification of the potential screens that should be combined, for example because they share the same set of tested drugs. Also it will improve the choice of the cell lines set to maximize the accuracy of the prediction. In the future, our parallel screens prediction method could be scaled up on this large set of data.

Additionally, recent work [91] compared the performance of a deep learning classifier and an ensemble-based tree classifier for MOA prediction within and across cell lines. They showed that an ensemble-based tree classifier is generally boosted by the incorporation of additional cell lines, although not for all combinations of cell lines. More interestingly, this work reported that their convolutional neural network (CNN) sees its performance decreasing with the addition of further cell line data. Hence, for the task of improving target prediction while combining parallel cell lines, a random forest classifier seems more robust and generalizable.

Chapter 4

Cell spatial arrangement could bring further functional information for drug profiling.

4.1 BBBC021, a cornerstone dataset in High-Content Screening analysis

4.1.1 A High-Content Screening dataset with breast cancer cells

This result section relies entirely on the publicly available dataset *BBBC021*, hosted by the Broad Institute [201] and first published by Caie et al. [202]. MC7 breast cancer cells were cultured and treated with anti-cancer drugs at different concentrations in 96-well plates for 24 hours. The general pipeline can be seen on Figure 4.1. They are then stained with DAPI and and immunostained against actin filaments and tubulin.

4.1.2 A benchmark of profiling methods on the BBBC021 dataset

My work on this dataset was based on the benchmark of profiling methods presented in the analysis by Ljosa et al. [34] They worked on a reduced set of images from the BBBC021 dataset, from which features were extracted using CellProfiler [26]. The drug and concentration conditions Ljosa and collaborators used [34] and which I used too, can be visualized on Figure 4.2. Drugs are not represented by the same number of cells because there are different numbers of concentrations per drugs and different numbers of cells per image due to the relative toxicity of the drug.

Ljosa and coworkers tested simple and more complex profiling methods already used and published in High-Content Screening (HCS) analyses. Their main conclusion was that the simplest method, averaging feature values element-wise per treatment, is as good

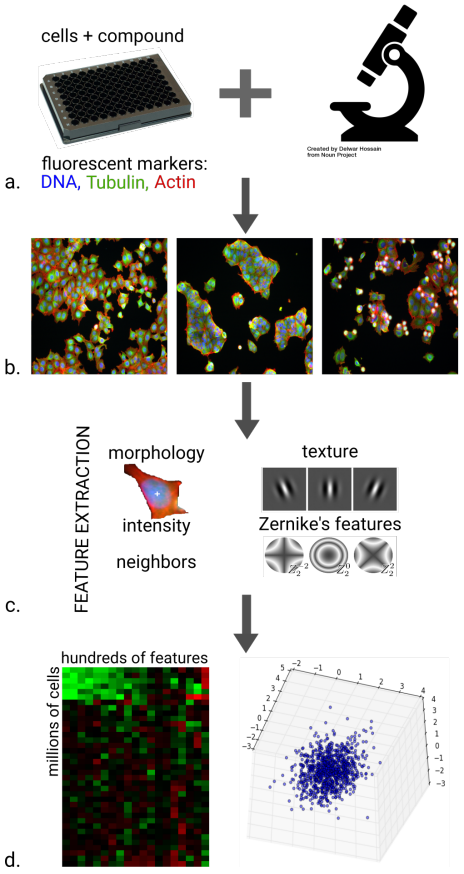


Figure 4.1: **Experimental and data extraction pipeline. a.** Cells are seeded on multiple well plates, then treated with DMSO (negative control) or a drug at a certain concentration. After 24 hours, cells are fixed and stained with three fluorophores marking DNA, Actin and Tubulin. Plates are fed to an automated microscope and **b.** four pictures per well were acquired. Cells are detected and segmented with CellProfiler. **c.** On each cell, features corresponding to the morphology, fluorescence intensities, texture and neighbors are computed. **d.** These features can be visualized under the form of a matrix where each row correspond to a cell or points in an high-dimensional space where each point represents a cell.

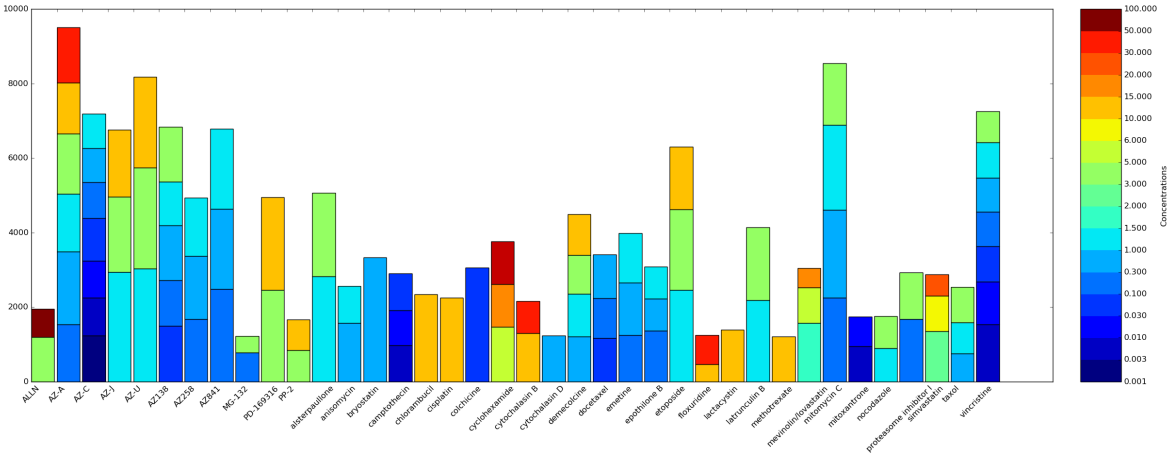


Figure 4.2: **Number of detected cells on images per treatment.** The x-labels correspond to drugs. The colors correspond to concentrations. The y-axis represents the number of detected cells.

as more complex methods like Gaussian Mixture Model (GMM) or Support Vector Machine (SVM). To compare profiling methods, they used mechanism of action (MOA) annotations: the 38 drugs were classified in 13 MOAs.

They followed a few steps to construct confusion matrices and global accuracy (cf Figure 2.6 page 39):

- Each profiling method provides a feature vector per well.
- The median vector between the biological replicates is computed, outputting a feature vector per treatment. The biological replicates correspond to different wells treated with the same compound at the same concentration.
- For each treatment X, the dataset is separated into two groups: the treatments with other concentrations of the drug used for treatment X and the rest.
- In the rest of the dataset, pairwise distances between the treatment X and the other treatments are computed with the cosine metric.
- The closest treatment to X is selected in the rest of the dataset, and its MOA is chosen to be the one predicted MOA for treatment X.
- The confusion matrix is completed according to the real MOA of X and the predicted one.
- The global accuracy is computed as the sum of the diagonal over the sum of the entire confusion matrix.

This methodology can be called leave-one-drug-out cross validation, with nearest neighbor voting.

Ljosa and coworkers proposed factor analysis as a profiling method, which is scoring higher for accuracy (94% compared to 83% for the features' mean method), however they tried several number of factors to get the best fit for this dataset. This parameter tweaking makes it impossible to get the best results from this method in a real setup.

With the images, features and MOA annotations publicly available, the BBBC021 dataset has been widely used to test and compare image analysis or

profiling methods [31, 33, 34, 78, 89, 90, 203–205].

4.1.3 Features' extraction and segmentation

In this chapter, some results presented use directly the features extracted by Ljosa and collaborators, however for the later part, as I needed the full segmentation and label images, I made a new segmentation, following a pipeline comparable to the one used by Ljosa and coworkers [34, 202]. Using the CellProfiler software, the pipeline comprises a few steps.

First illumination images per channel are computed using all the images from the screen, then used to correct the illumination defaults.

Then a nuclear segmentation step is applied based on a combination of pixel contrast and high intensity values in the nucleus channel. Declumping and reshaping steps are used to improve the nuclei segmentation: an iterative setting of shape and intensity metrics works to separate clumped nuclei and oddly shaped nuclei across compound-induced perturbations.

Afterwards, the cytoplasm for each detected nucleus is segmented via an object-based method: a watershed algorithm is applied with each nucleus as seed and using the intensity of the actin channel.

All the size, shape, intensity, texture and neighbor features available in CellProfiler are extracted for the three defined regions-of-interest: nucleus, cytoplasm, and whole cell. Due to compatibility of CellProfiler versions and Linux operating system, I moved the pipeline to CellProfiler3. Although the number of available features vary from version to version, the core features remain the same.

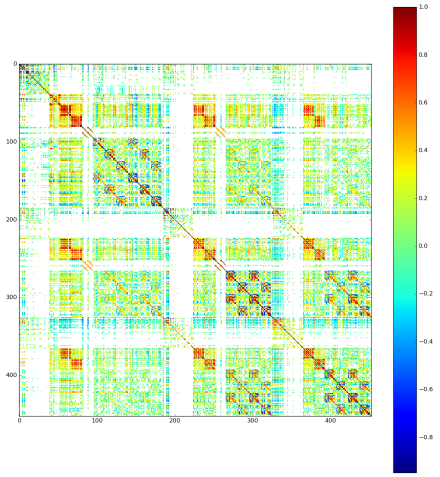


Figure 4.3: **Matrix of the pairwise Pearson correlation coefficients between the 453 features.** Features are taken from [34].

4.2 Structure of the feature space

4.2.1 Correlation of the features

The idea here is to have some metrics to see how Ljosa's features are affected by the curse of dimensionality. One approach is to look at the correlation between features (cf Figure 4.3). On the correlation matrix, we can see groups of very correlated features. Features corresponding to different summary statistics of the intensity distribution are expectedly correlated, like the mean and the maximum for example. In the same way, different orders of the Zernike polynomials (texture features) are also correlated.

Feature selection based on the correlation matrix requires to give a meaning on the pairwise correlation values and to choose a meaningful threshold above which one of the two features has to be

removed. So it did not appear a suitable solution without any a priori on the wanted level of correlation or the number of features. The following question is about how the pairwise feature correlation affect the data and clustering methods (see Section 4.3.1.)

4.2.2 High-dimension's effect on distance distributions

When a query cell and its nearest neighbors according to the Euclidean distance in the feature space are considered, a strong resemblance can be observed (cf cropped images on Figure 4.4), leading to the hypothesis that small distances are still meaningful even in this high-dimensional space. When the distribution of these nearest neighbors' distances to the query cell (19 NN in the figure) is visualized via a boxplot (cf upper part of Figure 4.4), these distributions are very narrow, but of different mean values for different query cell. So the 19 nearest neighbors of a point are approximately at the same distance, compared to the range of distance values existing in the dataset.

This distance is characteristic to the query point, and can be correlated to its distance to the data cloud center. Indeed, when plotting the 2D distribution estimation of the distance to the mean point of the data cloud against the distance to the kth NN, there is a strict linear correlation (cf Figure 4.5, non-shuffled, Euclidean metric, k=10, Pearson correlation p-val=0.0, corr = 0.88).

To conclude, there seem to be a conserved meaning in small distances, as cells at a small distance of a query cell is likely to look alike, however the value of its distance cannot be taken as a raw quantity and cannot be transposed in another part of the high-dimensional space to quantify how alike are two random cells.

4.2.3 K-nearest neighbors distribution

Another interesting phenomenon is the hubness which counts the number of times a point appears among the kNN of all the other points in the dataset [121]. It is thought that when increasing the dimensionality (see Section 2.2.5), the distribution of the hubness becomes skewed to the right: most of the points will not be in any of the kNN lists of other points, and some points will be present in all kNN lists. The skewness of the hubness distribution is defined as followed:

$$S_{N_k} = \frac{E(N_k - \mu_{N_k})^3}{\sigma_{N_k}^3}$$

with N_k the hubness for k nearest neighbors, μ_{N_k} the mean of the hubness distribution, and σ_{N_k} the standard deviation of the hubness distribution.

If $S_{N_k} = 0$, there is no skewness, the distribution is symmetrical on both sides of the mean, however if $S_{N_k} > 0$, there is skewness to the right of the mean, and if $S_{N_k} < 0$, skewness to the left of the mean. The value increases with the intensity of the phenomenon.

Hubness is caused by some points that are by chance closer to the mean of the point cloud than the majority of points, and these are likely to become hubs [121].

I computed the skewness of the hubness for the BBBC021 dataset with both Euclidean and cosine metrics. Results can be seen on the Figure 4.6: the skewness has a value of 3.02 for cosine metric and for 10NN, and of 3.72 for Euclidean metric and 10NN. The skewness is slightly higher for Euclidean distance.

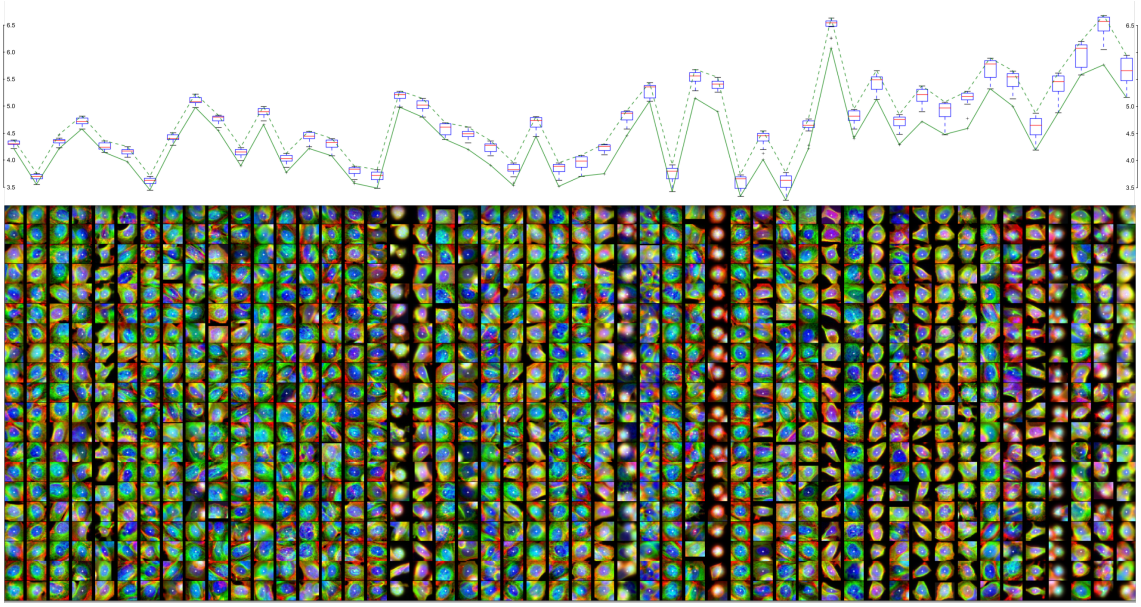


Figure 4.4: **Boxplot of the 19 nearest neighbors distances in the feature space from a random query cell.** Each query cell is the image on the first row, followed by its 19 nearest neighbors in column. Euclidean distance was used.

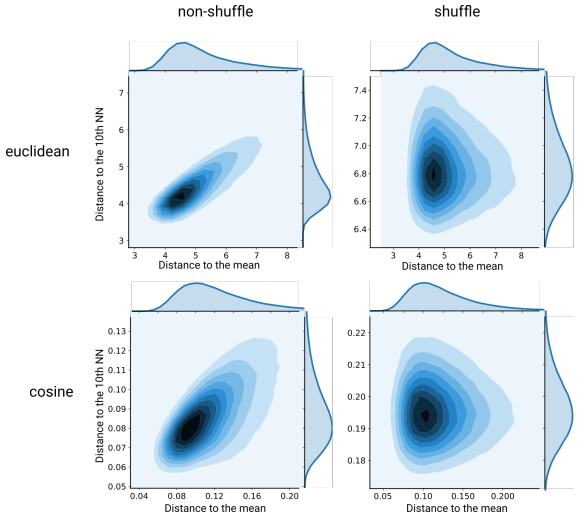


Figure 4.5: **Density of cells according to their distance to the data cloud mean and to their 10th NN.** For each cell, the distances to the data cloud center and to its 10th nearest neighboring cell are computed. Euclidean and cosine distance metrics were tested. Non-shuffle label corresponds to the real data, shuffle label to a shuffling of values between the features, removing the correlation seen between features.

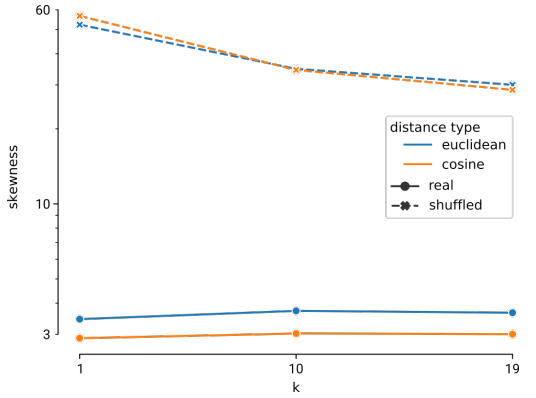


Figure 4.6: **Skewness values for different values of k-nearest neighbors**, for the real data or shuffled data, and for Euclidean and cosine metrics. Note the log-scale used for the y axis.

For Euclidean distance and independent and identically distributed (i.i.d.) uniform data, the skewness equals 1.54 for 20 dimensions, and 5.45 for 100 dimensions. Hence our dataset which lives in a 453-dimensional space, has a skewness smaller than a dataset in a 100-dimensional space that has completely independant dimensions. I also shuffled the values of all points (148 596 points) per feature: as a consequence, there is no correlation between the features anymore, and I recomputed the skewness values of the hubness. The skewness

values for the shuffled data are much higher, with a value of 34.44 for cosine metric and 34.81 for Euclidean metric. So the real data have a much lower skewness than expected from its dimensionality, resulting from the fact that the data are not completely random and from the correlation between features.

4.3 Detecting subpopulations in high-dimension

4.3.1 Benchmarked clustering methods

Motivation

We decided to focus on clustering methods because we wanted to have access to subpopulations of cells in an unbiased way. Supervised methods are well purposed to learn boundaries of absolute categories, defined by metadata like treatments or human labels of easily defined objects like tables or cars. Differently, subpopulations are loosely defined: there are no objective metadata on each of the cell, and even expert biologists cannot surely annotate the dataset with more than a few labels corresponding to obvious phenotypes like punctuated actin, round, fragmented nuclei, (de)stabilized tubulin [202]. Clustering methods use the structure of data and similarities between points to identify subgroups.

Tested methods

With features provided by Ljosa et al. [34], I tested clustering methods available in the *Python* package scikit-learn: spectral analysis, k-means and DBSCAN [74]. However DBSCAN and spectral analysis were not tractable for such an amount of data (cf Figure 4.7).

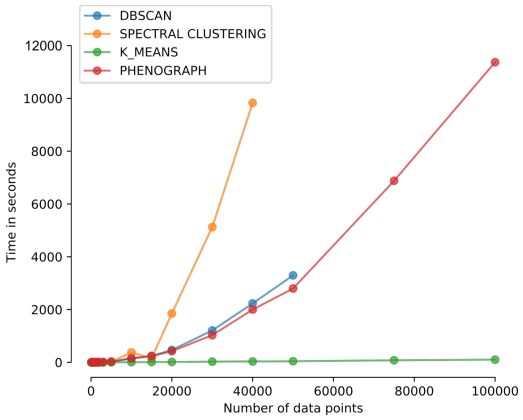


Figure 4.7: **Computing time of 4 clustering methods in function of the number of data points.**

Also DBSCAN, even with varying parameters could not find several cluster, but a single cluster and outliers. From published biology-driven methods, I tested two methods developed for cytometry data: PhenoGraph [116] and flowSOM [52]. I also tested a method for HCS data published in 2010 by Ng and coworkers [94]. The Ng method comprises a succession of steps, including feature selection and k-means per channel.

Methodology

On the same data matrix, I tested the following methods for subpopulation extraction:

- k-means with 10, 15, 20, 25 clusters;
- k-means per channel, 3 to 5 clusters per channel, later concatenated in one label (extract from the method used by Ng and collaborators [94]), making 8, 27, 64 clusters;
- PhenoGraph with $k \in [5, 10, 15, 20]$, k being the parameter to build the kNN graph;
- flowSOM with square grids of edge 5, 7, 10, creating 25, 49, or 100 clusters. I did not use the second clus-

tering proposed by flowSOM because it always aggregated the vast majority of cells in one cluster, loosing the advantage of splitting the data in several clusters.

I used 6 data matrices, combining two parameters: the number of cells, and the number of features. For the number of cells, I used:

- the full dataset (148 649 cells) of cells from non-DMSO images
- a restricted dataset (146 126 cells) of cells from non-DMSO images, taken randomly with a maximum of 1,000 cells per image to limit oversampling of some categories.

For the number of features, I used:

- all of the 453 features used by Ljosa and coworkers [34],
- 253 features selected by the feature selection described in the work of Ng and collaborators with the Pearson threshold set at 0.85, and the p-value threshold for the Kolmogorov-Smirnov (KS) test at 0.05.
- 239 features, corresponding to the same set of 253 selected features without the features relative to the neighbors, such as the percentage of touching membranes, or the distance to the closest neighboring cell.

I evaluated each method on the six variations of the dataset with two criteria: the accuracy of MOA prediction as computed by Ljosa and collaborators [34] (see Section 4.1.2), and the replicate similarity. The replicate similarity is computed as $1 - overlap$, with *overlap* being the overlap between the two following histograms: the histogram of pairwise distances between wells that correspond to replicates, and the histogram of pairwise distances between wells that are not biological replicates of each other. The two histograms are normalized before the overlap is computed.

Each well is described by the proportion of cells member of each subpopulation, it constitutes a numerical vector summing to 1. A median vector per treatment is computed from the biological replicates to compute the MOA accuracy.

Results

Results from the benchmark of clustering methods are summarized in the Figure 4.8. On panel **a.**, the range of accuracies can be observed for the four methods, flowSOM, k-means, k-means per channel and PhenoGraph, for the 3 sets of features. k-means per channel performs poorly with lower accuracies than the other methods and displays a higher variability for the distribution of accuracies. flowSOM and PhenoGraph have comparable accuracy results and distribution variability. k-means performs as well as flowSOM and PhenoGraph when the dataset comprises 453 features, but this performance decreases when features are removed from the original pool.

On panel **b.**, each run of a clustering algorithm on one dataset with one set of parameters is displayed with one dot. The size of the dot represents the number of features. In this representation, it is easy to see that k-means per channel performs worse than the other methods. flowSOM achieves similar accuracies as PhenoGraph or k-means but with a higher number of clusters. k-means

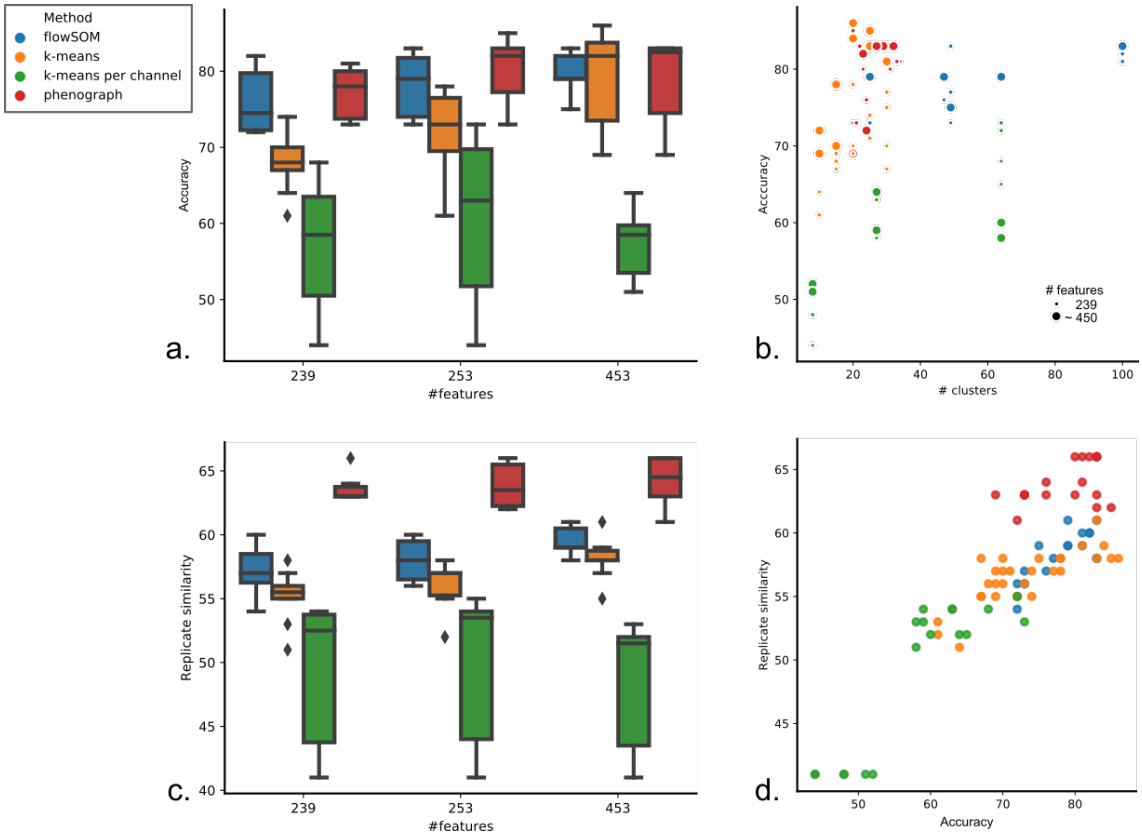


Figure 4.8: **Comparison of accuracy and replicate results for benchmarked clustering methods. a.** MOA accuracy following the LOOCV method from [34]. X-axis represents the number of features in the dataset. The distribution is computed from values obtained with the two different matrices and different sets of parameters. The color codes for the method. **b.** Accuracy versus number of clusters. The size of the points represent the number of features. Same colors for the methods. Each point is one run of a clustering with a certain parameter set and on one of the matrices. **c.** Replicate similarity in function of the number of features in the tested dataset. **d.** Replicate similarity versus accuracy. The color signals for the method.

achieves the highest accuracies but also average ones, around 60-65%. Visually there is no relation between the number of features and the accuracy performance.

The panel **c.** bears the same representation as panel **a.** with replicate similarity instead of accuracy. Once again, k-means per channel performs poorly. However, flowSOM and k-means perform equally with replicate similarities around 55-60%, and PhenoGraph performs the best with values above 60%.

On panel **d.**, each clustering run is represented by a dot for its accuracy and its replicate similarity. There is a loose linear correlation between the accuracy and the replicate similarity. Yet, we see a clear grouping of points per clustering methods, with PhenoGraph (red dots) at the top with high replicate similarity and high accuracy. I have not found differences between the two sets of 144 126 samples and 148 649 samples.

The results of a clustering run can also be observed by looking directly at the cells grouped together, as displayed in Figure 4.9. For this run of PhenoGraph, we can see that most of cells belong to a few big clusters, and the last clusters (17 and 18) correspond to false detections. Visually the grouping is satisfying. On Figure 4.10, the confusion matrix corresponding to the same PhenoGraph clustering as the one used for Figure 4.9 is displayed. Most of the data points fall on the diagonal, however there is some confusion between DNA damage and replication on one side, and protein degradation and synthesis on the other side.

Conclusion

Based on the accuracy criterion, flowSOM, k-means and PhenoGraph are good choices. flowSOM requires an important number of clusters, which is a bit problematic for our purpose: indeed, our goal is to compare different treatments, and many clusters will then have a low number of cells assigned to them in only a few treatments. k-means have a higher variability with respect to the input parameter k which sets the number of clusters. It has a lower replicate similarity than PhenoGraph as well. Phenograph is an interesting method because it is based on the k-nearest neighbor graph, which keeps only the very small distances. We saw that small distances seemed to preserve a biological meaning, as nearest neighbors of query cells in the feature space resemble the query cell (cf Figure 4.4). Secondly it uses an objective measure, the modularity (cf Section 2.2.4), to choose the number of output clusters.

4.3.2 Grid-search parameters for PhenoGraph clustering method

I chose to study the impact of input parameters on the results of PhenoGraph because it was found to be a promising method for clustering the BBBC021 dataset into cellular subpopulations according to features extracted by Ljosa et al [34].

Methodology

For the PhenoGraph method, I search the best couple of parameters $(k, min\_cluster\_size)$, with k the number of neighbor connections for each cell in the kNN graph, and

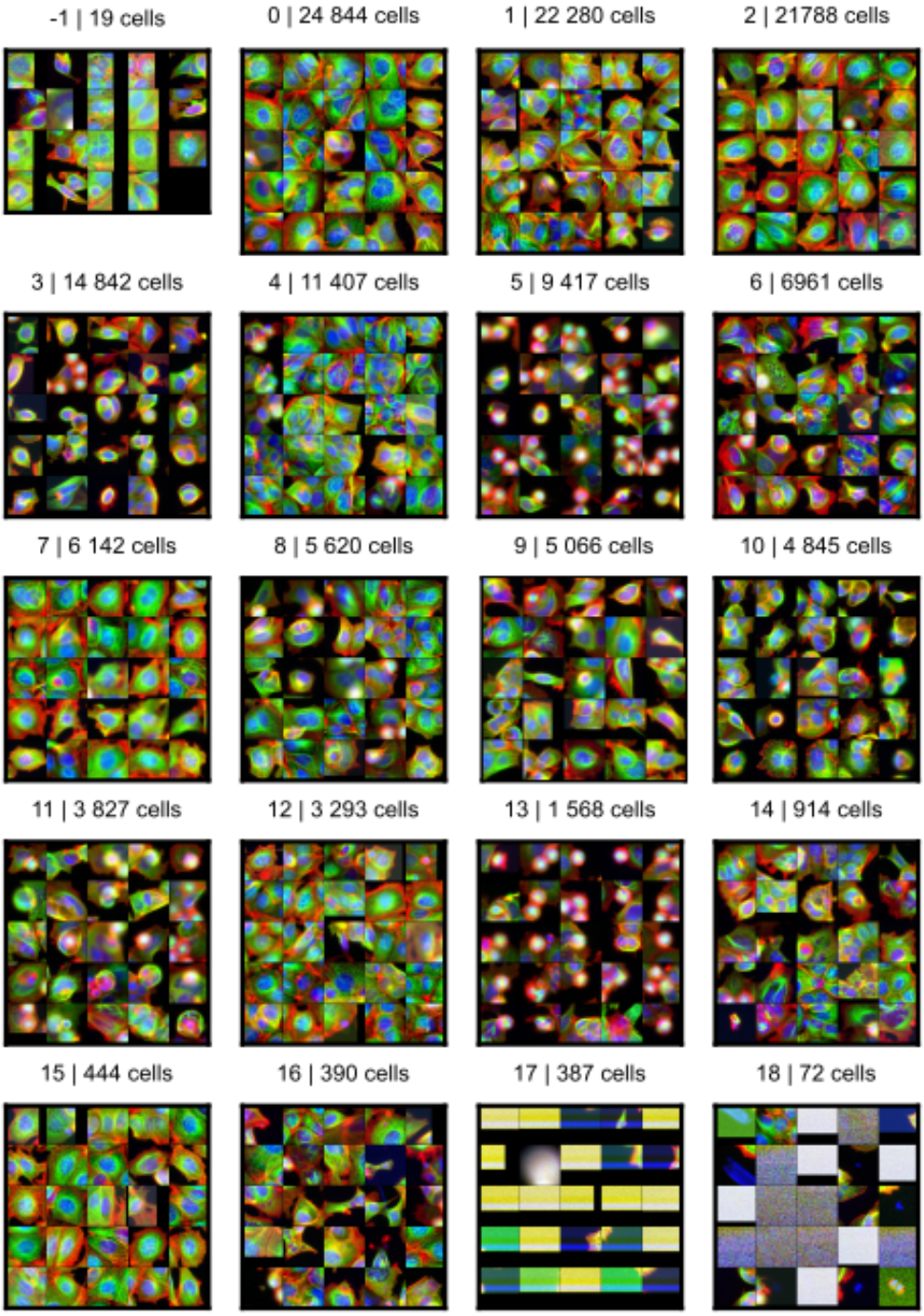


Figure 4.9: **Random cropped cell images from PhenoGraph clusters.** The first group of cells correspond to the label -1 and are outliers, i.e. are not part of any cluster. Then the clusters are labeled from 0 to 18, in the decreasing size order. For each cluster, the number of member cells is mentioned on top of the cropped cell images. Phenograph clustering obtained with the data matrix, containing 144,126 samples and 239 features. $k = 15$.

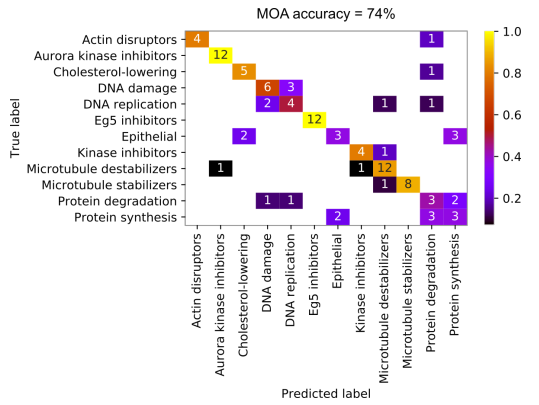


Figure 4.10: **Confusion matrix for MOA prediction.** For each treatment, the other treatments corresponding to the same drug are put aside and the closest treatment according to the PhenoGraph subpopulation proportions is computed with a cosine distance. The predicted MOA is the one of the closest treatment. The corresponding square in the matrix is incremented as necessary. The global MOA prediction accuracy is computed as the fraction of the sum of the diagonal on the total sum of confusion matrix. The colormap corresponds to the relative accuracies per line. Phenograph clustering obtained with the data matrix, containing 144,126 samples and 239 features. $k = 15$.

min_cluster_size as the minimal number of points which are required to form a cluster. The tested parameter values were for k, $2, 3, 5, 10, 15, 20, 30$ and for *min_cluster_size*, $5, 7, 10, 15, 20, 50$.

I evaluated the clustering outcomes on several criteria: the number of clusters, the average size of clusters, the standard deviation of the cluster size, the size of the biggest cluster, the number of clusters below 100 data points, the number of outliers (points not assigned to any cluster, but assigned the label -1), the MOA accuracy and the replicate similarity as defined above.

The clusterings were tested on a data matrix of 139 058 samples and 416 features not comprising the neighboring features, from the segmentation made with CellProfiler3. I focus on a feature set not including the neighboring features, as the rest of my work is using other means to study the neighboring relationships between cells (see Sections 4.4 and 4.5).

Results

Results are summarized in the Figures 4.11 and 4.12. For most of the criteria, changing the value of the *min_cluster_size* parameter has very low impact. However, changing the value of k changes drastically the distribution of the number of clusters, the number of outliers, the accuracy and the replicate similarity. For values of $k \in \{2, 3, 5\}$, there are many small clusters, and at least half of the cells classified as outliers. The accuracy for $k \in \{2, 3\}$ is below 65%, and the replicate similarity below 50%. Altogether, the values of $k \in \{2, 3, 5\}$ are not suitable to classify our cells into subpopulations as most of cells stay outside of clusters and the classification performance evaluations are poor.

For the criteria corresponding to the cluster size distribution, the trend continues as the value of k increases: the number of cluster decreases as the cluster size augments in average, and the number of outliers falls close to 0. On the other hand, for criteria like accuracy and replicate proximity, there is no global trend. Accuracies are between 80 and 90%, replicate proximities between 70 and 78%.

Conclusion

The best parameter set for accuracy is $(15, 15)$, with an accuracy of 90% and a replicate similarity of 74%, this clustering groups cells in 33 subpopulations. The best parameter set for replicate similarity is $(10, 7)$ with an accuracy of 86% and a replicate similarity of 78%, this clustering groups cells in 43 subpopulations. However, with k above 10, most of combinations provide suitable results. Hence, PhenoGraph seems quite robust

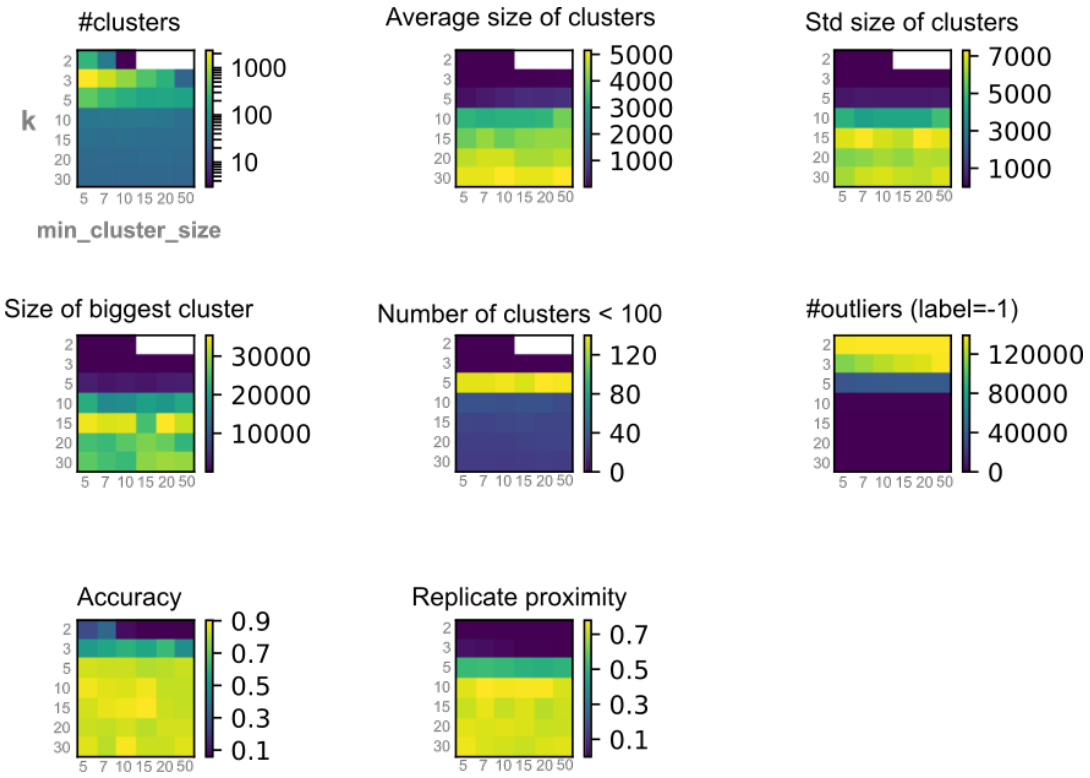


Figure 4.11: **Grid-search for input parameters k and $min\_cluster\_size$.** Each matrix displays different values of k on its rows and $min\_cluster\_size$ on its columns. Each subplot has its own color scale, depending on the nature of the displayed value. # is short for number, std for standard deviation.

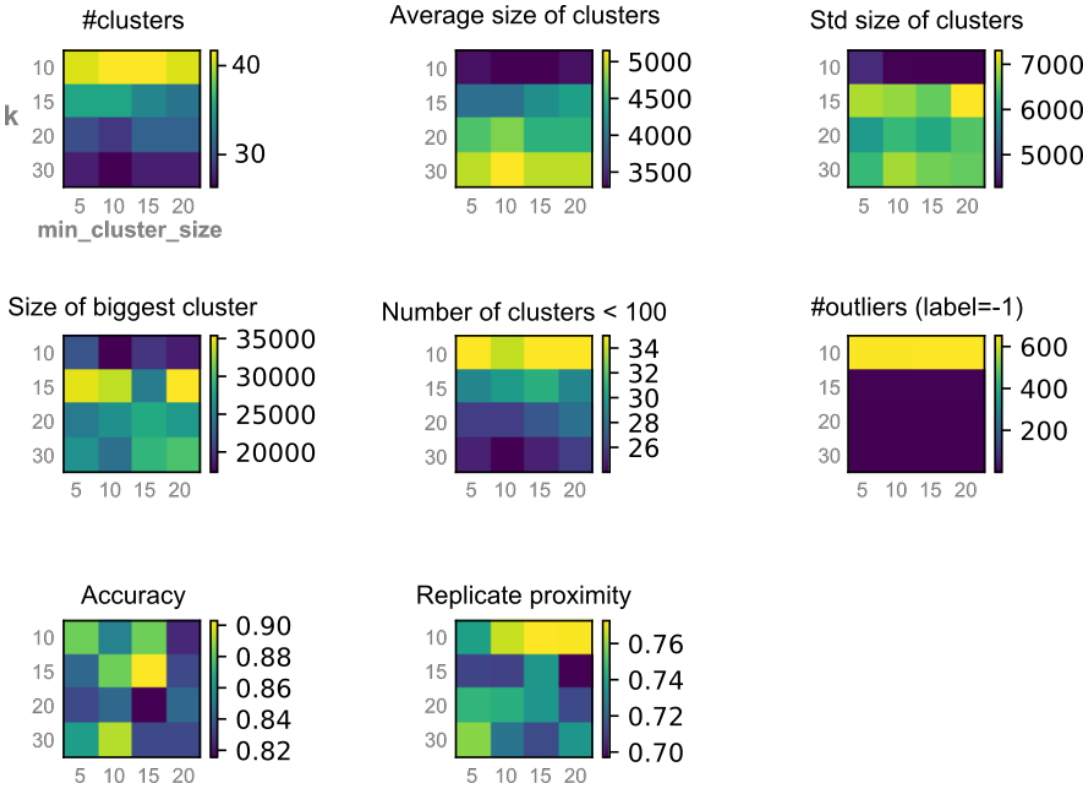


Figure 4.12: **Grid-search for input parameters k and $min\_cluster\_size$ (zoom).** Values displayed here are a subset ($k > 10$) of the ones from the Figure 4.11. Each matrix displays different values of k on its rows and $min\_cluster\_size$ on its columns. Each subplot has its own color scale, depending on the nature of the displayed value. # is short for number, std for standard deviation.

to changes in the input parameter values, and the overall quality of the clustering can be monitored with the number of outliers or the number of clusters (values of k too small).

4.3.3 Reproducibility analysis

Context

The above clustering methods are non deterministic. Due to the complexity of the data, in terms of number of samples and of number of dimensions, it is not tractable to explore the space of possibilities, hence the use of heuristics. For k-means, it depends on the initialization of the cluster centers. For PhenoGraph, the order in which points are considered while performing the cluster assignment matters and can influence the results. For flowSOM, in the same way, the initialization of the neurons to random data points as well as the random order in which points are fed to the network during the training step can influence the clustering [206]. These sources of instability are due to the fact, that before making groups, there is no obvious way to know where they are approximately lying.

Clustering similarity measures

I computed 20 runs of the clustering algorithms on the same matrix data with 144,126 samples and 239 features. Then I computed some indices to evaluate the similarity between pairs of outputs. These indices are the adjusted mutual information score, the v-measure, the adjusted rand score. They are all available in the scikit-learn *Python* package [74].

The mutual information of two clusterings U and V is equal to:

$$MI(U,V) = \sum_{i=1}^{|U|}\sum_{j=1}^{|V|} \frac{U_i \cap V_j}{N} log\big(N\frac{|U_i \cap V_j|}{|U_i||V_j|}\big)$$

with $|U_i|$ the number of samples in clusters U_i and N the total number of samples. This metric is symmetric, i.e. swapping U and V would not change the value of the mutual information.

The adjusted rand score is a normalized version of the rand index R:

$$R = \frac{a+b}{\binom{N}{2}}$$

with a the number of pairs of data points that are in the same cluster in U and also in V, and b the number of pairs of data points in different clusters in U and also in V. This measure is also symmetric.

The v-measure is the harmonic mean between homogeneity and completeness: $v = \frac{2*(homogeneity*completeness)}{(homogeneity+completeness)}$. It is a way to make the measure symmetric from two non-symmetric measures: homogeneity and completeness. Homogeneity checks if the points that are in the same cluster for the predicted labels belong to the same cluster in ground truth. Homogeneity equals 1 when this is strictly the case, and can decrease down to 0. Completeness is its mirror property: it is satisfied when all the points in the same cluster in the ground truth are elements of the same cluster in the predicted clustering.

As I don't have ground truth clusters, but I compare two predictions of the same clustering algorithm, I chose only symmetric measures.

Results

The three tested measures are highly linearly correlated (cf Figure 4.13 b.). On

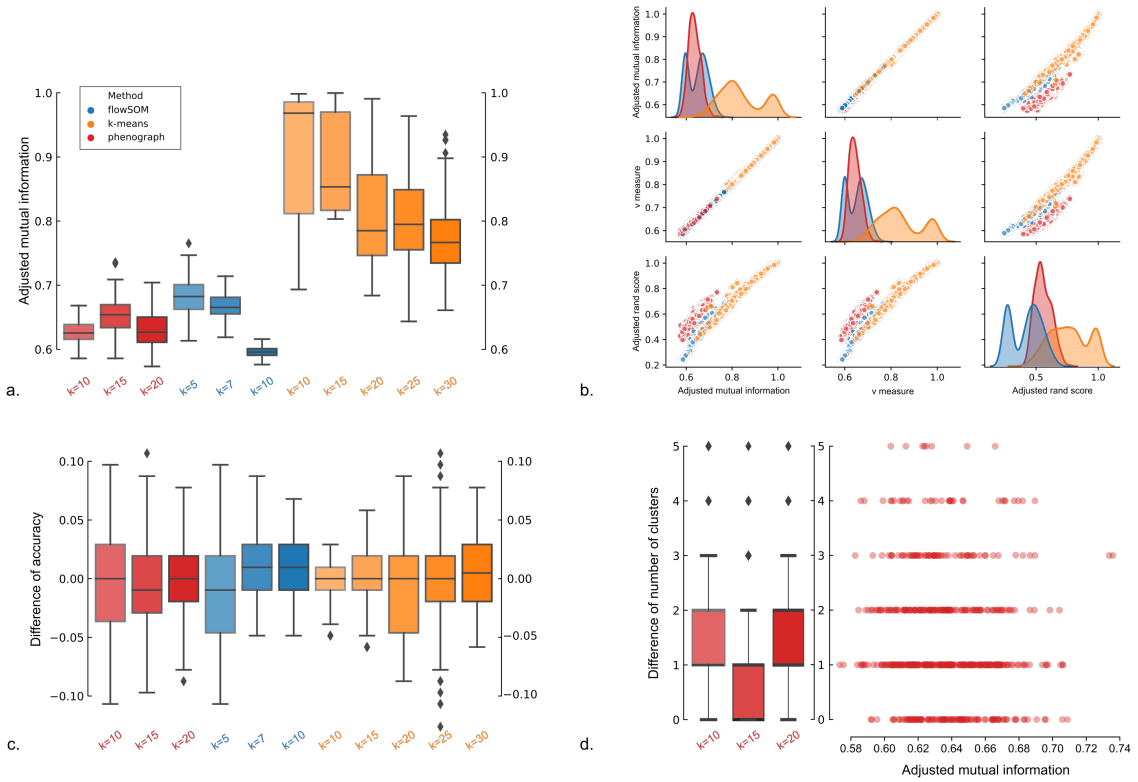


Figure 4.13: **Reproducibility results for k-means, flowSOM and PhenoGraph on 20 runs. a.** Mutual information criterion distribution for the 3 clustering methods, with key input parameter k. For PhenoGraph, k is the number of kNN for the proximity graph. For flowSOM, k is the size of the side of the square 2D neuron map (see section 2), the number of clusters is then k^2. For k-means, k is the number of clusters. Mutual information is computed between 2 different runs of the same clustering algorithm with the same input parameter. **b.** Pairplot of the 3 clustering similarity measures used. The scatter plots display the correlation between pairs of similarity measures. Plots on the diagonal are showing the 1D distribution for each similarity measure. **c.** Difference of MOA classification accuracy distribution between pairs of clustering runs for the 3 clustering methods, with key input parameter k. **d.** On the left, boxplot of the difference of the cluster number between pairs of PhenoGraph runs, with key input parameter k. On the right, adjusted mutual information between pairs of PhenoGraph runs against the difference in number of clusters.

the basis of 20 runs of PhenoGraph, with $k = 15$, i.e. 190 pairs of runs, we obtain an average value of 0.65 for mutual information, 0.61 for adjusted rand index and 0.66 for the v-measure. So the reproducibility between the 20 runs is intermediate (cf Figure 4.13 a.). FlowSOM has the same range of values. However k-means have higher mean values for all three measures: for parameter k ranging between 10 and 30, values are obtained between 0.64 and 0.99 (average 0.83) for adjusted mutual information score, between 0.40 and 0.99 (average 0.76) for adjusted rand index, and between 0.64 and 0.99 (average 0.84) for v-score as well.

The scikit-learn implementation of k-means uses the k-means++ initialization [207] which stabilizes and speeds up the computation, and this stable initialization could explain the higher similarities between runs.

On panel c. of the Figure 4.13, distributions of the difference of MOA classification accuracy between pairs of runs are displayed. These distributions have a low standard deviation, meaning that the between-runs variability of the clustering methods has little impact on the functional prediction. Some cells may not be put in the same groups, as mutual information and other criteria measure it, but the MOA classification is quite robust to these variations.

In the case of PhenoGraph, which is the sole method among the three that does not require the number of clusters as input parameter, the number of clusters can vary between runs. This variation is plotted on panel d. of Figure 4.13. The number of clusters can vary to a maximum difference of 5 for the tested input parameter values. No correlation between the difference in number of clusters and mutual information was observed (cf scatter plot on the right side of panel d.).

Conclusion

How to explain the lower performances of flowSOM and PhenoGraph compared to k-means? In opposite to the smart initialization of the k-means' algorithm, the fully random initializations of flowSOM and PhenoGraph can introduce variability in the results. Also, the data themselves may not clearly be separable, but it is hard to test it in high dimension. One observation supporting this is that several number of clusters achieve comparable functional prediction results. As for the results of PhenoGraph computation (cf Figure 4.9), most of the cells are members of a few big clusters, hence adding or removing some small clusters is not likely to impact the functional grouping and the functional MOA prediction while impairing the clustering similarity measures. Even with a very reproducible or deterministic algorithm, clustering is an ill-posed problem as we have no objective way to assess the performance in high dimension. Hence a reproducible clustering can still be far from the optimal solution.

4.3.4 Comparison of clustering methods and supervised methods

As clustering methods, I tested k-means, k-means per channel, spectral clustering, DBSCAN, PhenoGraph and flowSOM. Among those, DBSCAN and spectral clustering were not appropriate and did not give any result. From the exact same data matrix as Ljosa et al., the maximum reached accuracies are 64% for k-means per channel, 83% for PhenoGraph and flowSOM, and

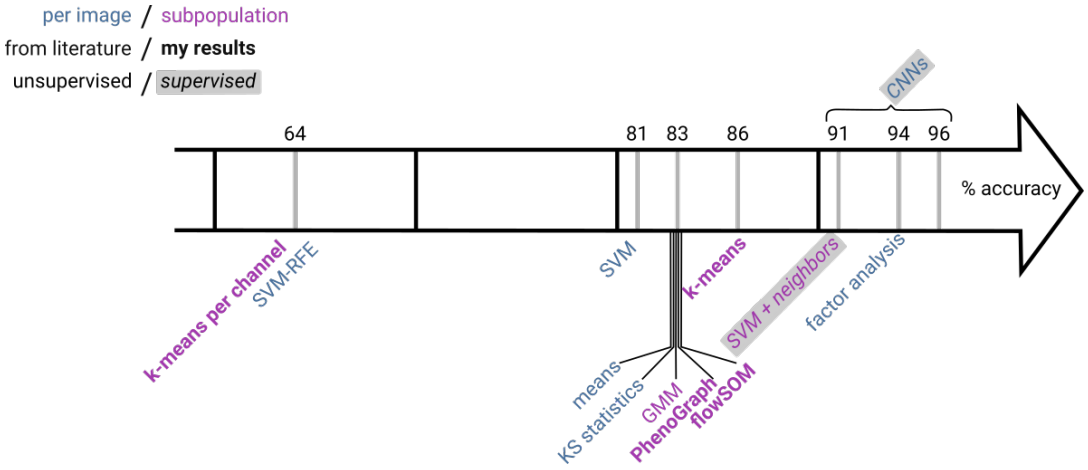


Figure 4.14: **Comparison of MOA accuracies from the literature and my results.** From left to right the scale of accuracy is expressed in percentage (103 treatments). The methods in purple correspond to subpopulation methods. The ones in blue deal with per image values. The methods in bold correspond to the methods I benchmarked. The ones on a gray block correspond to literature results with supervised methods, for CNN methods [31, 78, 89, 90, 203] and "SVM + neighbors" [204]. The rest corresponds to accuracies computed by Ljosa and collaborators [34].

86% for k-means. These percentages are added on the Figure 4.14, and can directly be compared to the ones published by Ljosa et al. [34]. In the methods tested by Ljosa and coworkers, all are unsupervised, but only one is looking for subpopulations: the Gaussian Mixture Model (GMM). For the non-subpopulations methods, all cells from one treatment are aggregated in one population. For example the KS statistic method computes the KS statistics on each feature between the distribution of the cells corresponding to one treatment and the cells corresponding to the negative control (DMSO). Accuracies from this article [34] are also added on the same figure (Figure 4.14).

The BBBC021 dataset is widely used across many research teams. And some other published articles developed and used diverse methods to recover the MOA classification. Among these, several methods use convolutional neural networks (CNNs), either with a training from scratch [31] or with transfer learning [78, 208]. Some feed to the network the original image or a large subpart of it [31, 89, 203], or feed directly cropped images centered on nuclei containing about one cell each [78]. They all achieve fairly good MOA classification accuracies, from 91% to 96%. These numbers can be compared with the ones previously detailed (cf Figure 4.14). However the comparison is less precise than between clustering methods based on the same exact data matrix as for each of these methods, the preprocessing, the neural network architecture, the learning method and other subtleties are assessed together when looking directly at the accuracy percentage.

If the only objective is to increase the MOA accuracy, then CNNs are the best overall option (cf Figure 4.14). However they usually do not provide information at the cell level, and do not allow to reconstruct subpopulations. They also require training data, as they belong to the supervised method category. Clustering methods to find supopulations have several advantages: they assign a label to each cell, allowing to describe the diversity of the biological responses in terms of subpopulations, and they do not require training data and cannot be biased by human-made categories. Ad-

ditionally on this dataset, they perform well.

Moreover, PhenoGraph is an interesting clustering algorithm in this situation. Indeed it is based on close relationships in the feature space, which can be trusted even in high-dimension. It is a method designed to be scalable to big networks. It does not require the number of clusters as an input parameter, which is a hard parameter to estimate in the case of cellular subpopulations. Based on the distribution of cluster sizes and the quantity of outliers, it is relatively easy to choose suitable input parameters as the number of kNN for the similarity graph construction and the minimal cluster size (cf Figures 4.11 & 4.12). It gives good accuracy results and best replicate similarity results (cf Figure 4.8). These are the reasons I used PhenoGraph clustering for most of the following work.

4.4 Observation of non-random spatial arrangement

4.4.1 Testing the first neighboring cell

Problematics

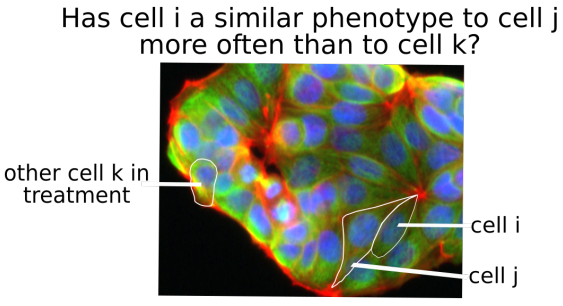


Figure 4.15: **Looking at spatial arrangement of subpopulations: "Has cell i a similar phenotype to cell j more often than to cell k?"**

While looking at the numerous images of this dataset, I got the impression that cells touching each other were looking more alike than expected when taking all cells present in the the image. I simplified the question to be able to test it (cf Figure 4.15). If three cells i, j, k of the same image are considered, such that cell j is the closest neighbor of cell i, and cell k is chosen randomly in the image, has cell i a similar phenotype to cell j more often than to cell k?

Methodology

I approximated the phenotype in this case by the cluster membership. I computed a difference of probabilities:

$$\Delta_C = \frac{1}{|C|}\sum_{i\in C}\left(1_{ph(i)=ph(j)} - 1_{ph(i)=ph(k)}\right)$$

with j the closest cell of i, k a randomly selected cell, and C the considered cluster. The cell k is chosen randomly, each cell c_l has then the probability $P(c_l) = \frac{1}{(N-1)}$ with N the total number of cells in the image, as cell i is removed from the pool. Hence:

$$\sum_{i\in C} 1_{ph(i)==ph(k)} = \sum_{i\in C}\sum_{l=1,c_l\neq i}^{N} 1_{ph(i)==ph(c_l)}P(c_l),$$

$$\sum_{i\in C} 1_{ph(i)==ph(k)} = \frac{|C|-1}{N-1}.$$

Then:

$$\Delta_C = \frac{1}{|C|}\sum_{i\in C} 1_{ph(i)==ph(j)} - \frac{|C|-1}{N-1}$$

For each image and for each cluster C, this difference of probability Δ_C is computed. For each cluster, the distribution of Δ_C per image can be plotted (cf Figure 4.16). $\Delta_C = 0$ means that the probability of having a neighboring cell member of the same cluster as itself is equal to the proportion of this cluster in the image. However $\Delta_C > 0$ means that

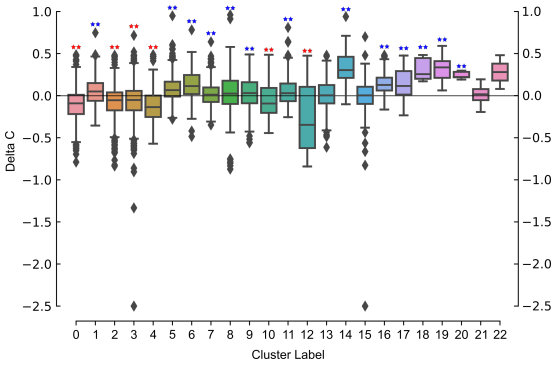


Figure 4.16: **Distribution of Δ_C per PhenoGraph cluster (x-axis).** One Δ_C value is computed per image and per cluster. A 2-sided t-test 1 sample was run for each per cluster distribution to test whether the mean is equal to 0. The significance of the t-test is indicated by stars: 1 star if $p - value < 0.05$, 2 stars if $p - value < 0.01$. The stars are red if the value of the t-test statistics in negative, indicating a mean lower than 0, blue otherwise.

the probability of two neighboring cells having the same cluster membership is greater than the proportion of the cluster in the image. To test if the distribution of Δ_C has a mean significantly different of 0, I performed a 2-sided t-test.

This was run on the same data as the first clustering comparison (cf Subsection 4.3.1): phenograph clustering obtained with the data matrix, containing 144,126 samples and 239 features. $k = 15$.

Results

On the Figure 4.16, we can see the boxplot representing the distribution of Δ_C values per image and per PhenoGraph cluster. The t-tests performed show that most of the clusters have a Δ_C distribution with a mean higher than 0 (13/22), even if some (6/22) have a mean significantly lower than 0. A mean higher than 0 means that cells of the same cluster are found next to each other more than expected from the proportion of cells from this cluster in the image.

However these results are partial because they are only based on the nearest cell neighbor: indeed for each cell only its nearest neighbor based on the Euclidean distance between their centroids is used. To take this analysis further, I decided to run a segmentation with *CellProfiler* based on the one Ljosa and collaborators published [34] to get access to all cell neighbors.

4.4.2 Testing all neighboring cells

Newman's assortativity

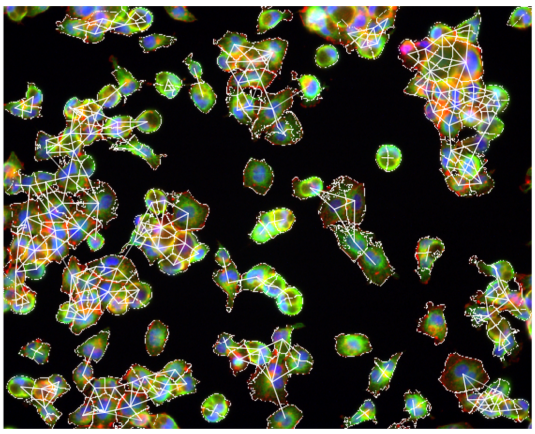
From the segmented image, where each pixel belonging to a cell i has the id i, the cell graph is easily reconstructed (cf Figure 2.7 page 45 from Section State of the art). From this graph (cf Figure 4.17 c.) which can be represented by an adjacency matrix, a counting matrix $\boldsymbol{e}$ is derived (cf Figure 4.17 b.), where the number of links between each type of node is reported. The counting matrix $\boldsymbol{e}$ is symmetric because the links between neighboring cells are undirected. Based on the counting matrix $\boldsymbol{e}$, the Newman's assortativity r is computed as described on Figure 4.17 d.

The results were obtained from the data matrix of 139 058 samples and 416 features not comprising the neighboring features, from the segmentation made with CellProfiler3 (cf Subsection 4.3.2), on which was performed a PhenoGraph clustering ($k = 20$).

Results

The distribution of Newman's assortativity values per image can be visualized on Figure 4.18 (blue histogram), with a mean of 0.09 and standard deviation of 0.08. According to a 2-sided

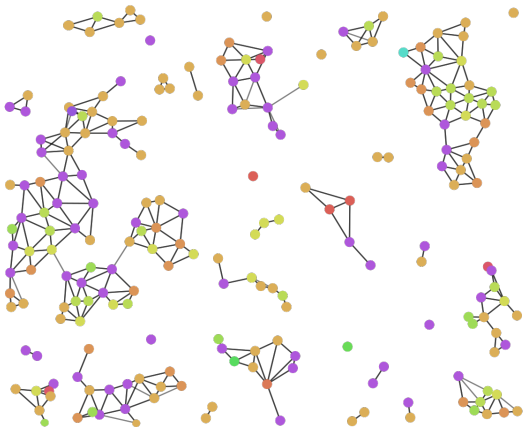
a.



b.

e_{ij}	0	3	...	18	a_i
0	1	2		2	5
3	2	27		25	54
...					
18	2	25		132	159
b_j	5	54		159	

c.



d.

$$r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i}$$

$$= \frac{\mathrm{Tr}\,\mathbf{e} - \| \mathbf{e}^2 \|}{1 - \| \mathbf{e}^2 \|},$$

Figure 4.17: **From the segmented image to Newman's assortativity. a.** Fluorescence image with overlay of the segmented cell borders and neighboring links. A link is put between cells if they share some membrane according to the segmentation. **b.** Counting matrix $\boldsymbol{e}$, intermediary step to compute Newman's assortativity. It counts how many links there are between two nodes with each label. The columns a_i and b_j correspond to the sum of links on the respective line or column. **c.** The graph corresponding to the image in a. Each dot represents a cell and is placed at its centroid, and the color of the dot represents the label given by the clustering. Neighboring links are drawn between cells that share membrane, according to the segmentation image. **d.** Newman's assortativity formula, computed with the numbers from the counting matrix $\boldsymbol{e}$. Tr means the trace of the matrix, $||\boldsymbol{e}^2||$ its norm.

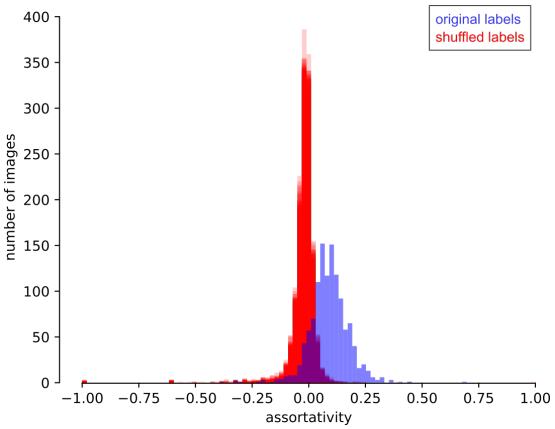


Figure 4.18: **Distribution of Newman's assortativity values per image.** The blue histogram corresponds to values of assortativity computed per image (cf Figure 4.17). The red histograms correspond to the same distribution but computed on images where the node labels were shuffled. 50 rounds of shuffling were done, each giving one red histogram. The shuffled distribution are very close and the corresponding histograms overlap largely. However on some bars, the differences of transparency show the variations between the round of shuffling.

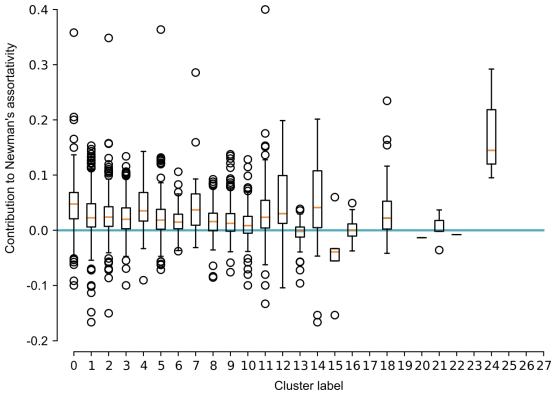


Figure 4.19: **Distribution of contribution to Newman's assortativity per cluster label.** For each image and for each cluster, the median of the r_i values was computed.

t-test, the assortativity distribution has a mean significantly greater than 0 (t-test $stat = 36.93$, $p - value = 10^{-200}$, on 1206 images). The same distribution was computed after shuffling the cell labels inside each image keeping the same cell graph. 50 rounds of shuffling were performed. The corresponding distributions can be visualized in red on the Figure 4.18. Their mean equals -0.02 with a standard variation of 0.06. 2-sided t-tests on each shuffling indicate a mean significantly lower than 0 (t-test $statistics \in [-14.37, -10.11]$, $p - value \in [10^{-43}, 10^{-28}]$, 50 shufflings on 1206 images).

Newman's assortativity is higher for the real data than for the shuffled labels data, meaning neighboring cells are more likely to be of same subpopulations than expected at random.

Newman's assortativity can be decomposed into a sum of contributions from each node label category. Indeed the contribution from a category i is:

$$r_i = \frac{e_{ii} - a_i * b_i}{1 - \sum_i a_i b_i}$$

using the same notations as in Figure 4.17. I pooled the values from images belonging to the same treatment and computed the median. The distribution of the median values per treatment and by node label category can be visualized on the Figure 4.19. We can observe that most of the distributions have a higher mean than 0, especially for the first clusters, containing most of the cells. When PhenoGraph labels its clusters, it does so by putting first the cluster with the more cells, then by decreasing size. It looks like most of the clusters contribute equally to the global assortativity, as there is no cluster with contribution values much higher than others.

4.4.3 Conclusion

The two previous approaches showed that the detected subpopulations do not lie randomly on the images. A spatial pattern exists. The next idea was to try to use the spatial information to complement the one based on cell phenotypes alone for drug profiling.

4.5 Leveraging non-random spatial arrangement for drug profiling

4.5.1 Via cell graph features

Motivation

To combine information coming from the phenotype and from the cell graph, the first idea was to transform the information content contained in the graph into features which could be later concatenated with the CellProfiler features averaged per image or per treatment, or the proportions of each subpopulation per image.

Methodology

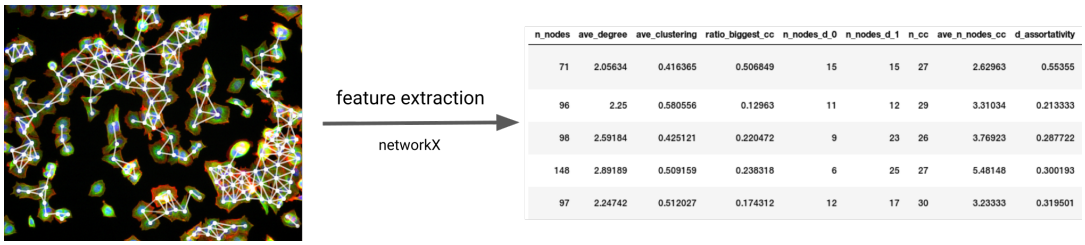
Graph features I used the NetworkX *Python* package [163] to extract available standard graph features, and also a set of features similar to the ones used by Yener and collaborators (cf Figure 4.20) [176]. Additionally, I added some tailored features such as the mean Euclidean cell distance between neighboring cells, the mean Euclidean cell distance between non-neighboring cells, the average convex hull size, and the average convex hull ratio. For each connected component in the graph, I computed the size of the corresponding convex hull, and the convex hull ratio (cf Figure 4.21) which is defined as:

$$convex\ hull\ ratio = \frac{\sum(area\ clique)}{area\ convex\ hull}$$

Normalization To control the variation between data coming from different plates, I applied a batch normalization step, inspired by the one used for the CellProfiler features [34]. On each of the plate, several wells of cells are treated only with DMSO, they correspond to the negative control. The distributions of features from cells treated with DMSO should coincide on every plate. The 1^{st} and the 99^{th} percentiles of each feature for DMSO-treated cells were put to 0 and 1 respectively. The intermediate values were scaled linearly between 0 and 1. The same linear transformation was then applied to all drug-treated wells of the same plate.

The difference between CellProfiler features and graph extracted features is that graph features are computed per image and not per cell. Hence the distributions of graph features corresponding to DMSO images have less data points. The transformation is then supported by a smaller amount of data. Moreover, some features like the number of nodes take different value ranges for DMSO-treated cells and drug-treated cells. Indeed in DMSO pictures, cells are more numerous as they are not submitted to a death-inducing treatment. Then the transformation function computed on DMSO images is not appropriate for other treatments, and sets general feature values not between 0 and 1, but within negative values. In this case, the normality of the feature distribution is not guarantied.

Concatenation I tested the graph features alone, but also in concatenation with CellProfiler features per image or PhenoGraph subpopulation proportions per image (cf Figure 4.22 a.). The features were pulled together per treatment, so for subpopulation proportions and graph features, the median of the values per image corresponding to the same treatment was taken. For CellProfiler features, they were first averaged by image, then the median was computed with the values corresponding to



n_nodes	ave_degree	ave_clustering	ratio_biggest_cc	n_nodes_d_0	n_nodes_d_1	n_cc	ave_n_nodes_cc	d_assortativity
71	2.05634	0.416365	0.506849	15	15	27	2.62963	0.55355
96	2.25	0.580556	0.12963	11	12	29	3.31034	0.213333
98	2.59184	0.425121	0.220472	9	23	26	3.76923	0.287722
148	2.89189	0.509159	0.238318	6	25	27	5.48148	0.300193
97	2.24742	0.512027	0.174312	12	17	30	3.23333	0.319501

Figure 4.20: **From the cell graph to the extracted graph features.** The cell graph is converted into a NetworkX object with only the node and links between nodes, without the coordinates of each node in the image or the labels. Features extracted with NetworkX were then mainly extracted from the node distribution statistics. Additional features taking into account the image space were manually designed. On this figure, only examples of these features are displayed.

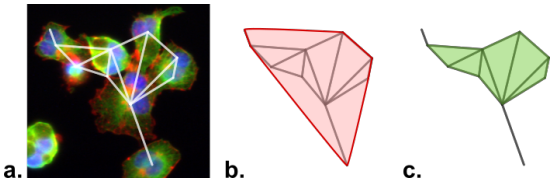


Figure 4.21: **Connected component of the cell graph, its convex hull and its convex hull ratio. a.** Crop of an image around one connected component of the cell graph. The graph links are overlaid in white. **b.** Convex hull drawn in red over the graph connected component. In pale red is the convex hull area. **c.** In pale green is the area corresponding to the sum of the areas of cliques of the connected component. The convex hull ratio is the ratio between the green and the red areas.

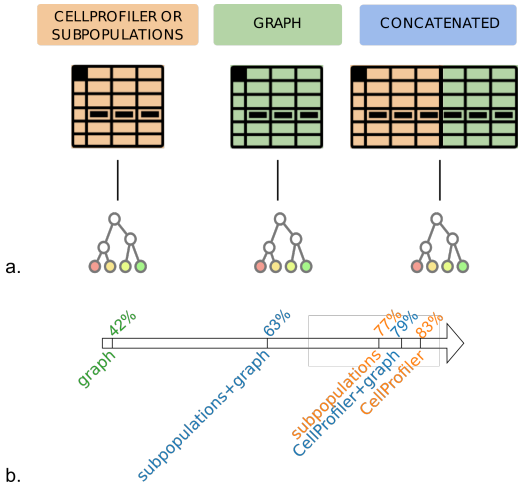


Figure 4.22: **Random forest MOA classification for graph, CellProfiler and subpopulations features. a.** Each matrix of data is given to a separate random forest classifier. **b.** The results in terms of MOA prediction accuracies can be compared.

the same treatment images. All the following results are computed from these median feature vectors per treatment.

MOA prediction For MOA prediction, two classification approaches were used: the unsupervised leave-one-compound-out cross validation (LOO-CV) with nearest neighbor (NN) voting (cf Section 4.1.2), and a supervised random forest (RF) method. I chose to supplement the first with a random forest because this method can select some features as more interesting for the classification, and discard some less interesting ones. It allows to have a rating of features for the MOA classification task and see the relative importance of the CellProfiler, subpopulation and graph features. For the random forest classifier, I also used the leave-one-drug-out cross-validation procedure. The confusion matrix is constructed per treatment and the global MOA accuracy computed.

Results

Each of the data matrices: graph features, CellProfiler features, subpopulation proportions, and the concatenated ones, graph&CellProfiler features and graph&subpopulation features, is fed separately to a RF classifier or to the NN voting procedure (cf Figure 4.22 a.). The average accuracies from the confusion matrices for all LOO-CV RFs are displayed on Figure 4.22 b. and Table 4.1.

	RF	NN
graph	42	43
subpopulation	77	80
CellProfiler	83	85
graph&subpopulation	63	45
graph&CellProfiler	79	74

Table 4.1: **Comparative results for global MOA accuracies per treatment with random forest or nearest neighbor predictions.** The three sets of features per treatment are graph features, CellProfiler features and subpopulation proportions, with the additional concatenated sets of features: subpopulation proportions and graph features, and CellProfiler and graph features. RF stands for Random Forest and NN for Nearest Neighbor voting. The numbers are global MOA accuracies expressed in %.

Random Forest results For the matrix with only graph features, the random forest classifier reached 42% accuracy, compared to 77% for the subpopulations and 83% for the CellProfiler. We see that the graph features have a much lower prediction power than subpopulations or CellProfiler features. For the concatenated matrices, the random forest accuracies are respectively 63% for subpopulation proportions and graph features, and 79% for CellProfiler and graph features together. The accuracies of the concatenated matrices are higher than the one from graph features alone, but lower than the ones from subpopulation proportions or CellProfiler features.

Random forest measures the features' importance by looking at how much the tree nodes related to one feature reduce impurity across all trees in the forest. The impurity is calculated via the Gini index. The node is pure if all examples falling under this node correspond to the same category. All the feature's importance values together sum to 1. As seen of Figure 4.23, the two most important features are the mean and the standard deviation of the cell-to-cell distances. Then most of the features have approximately the same importance. The cell-to-cell distance measure is closely related to the area of the

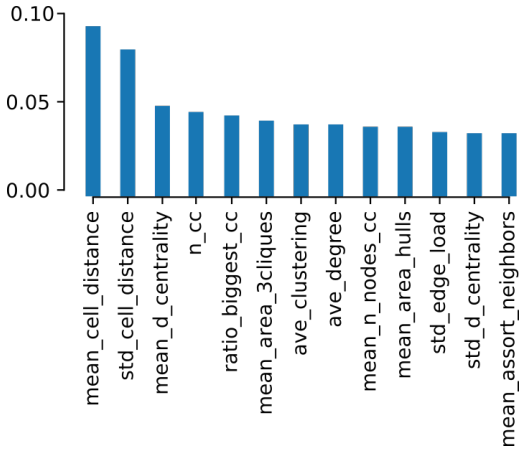


Figure 4.23: **Relative graph feature importance computed by the random forest algorithm.** The 13 first features are displayed in order of importance. The features are: *mean_cell_distance*, the mean cell-to-cell distance; *std_cell_distance*, the standard deviation of the cell-to-cell distances; *mean_d_centrality*, the mean degree centrality; *n_cc*, the number of connected components; *ratio_biggest_cc*, the ratio of the size of the biggest component over the size of the graph; *mean_area_3cliques*, the mean area of the 3 cliques; *ave_clustering*, the average clustering coefficient (the clustering coefficient of a node is the fraction of possible triangles through that node that exist); *ave_degree*, the average degree; *mean_n_nodes_cc*, the mean number of nodes inside each connected component; *mean_area_hulls*, the mean area of convex hulls; *std_edge_load*, the standard deviation of the edge load (the edge load is the number of paths passing through an edge); *std_d_centrality*, the standard deviation of the degree centrality; *mean_assort_neighbors*, the mean degree assortativity of the neighboring nodes.

cells, and is not fundamentally a new benefit from the graph feature extraction compared to CellProfiler features.

Nearest Neighbor voting results The MOA prediction accuracies for random forests are slightly lower than with the NN approach for the simple sets of features and higher for the concatenated ones (cf Table 4.1). More precisely, the drop in accuracy when adding the graph features is more important for the subpopulation proportions as it goes from 80% to 45%, than for the CellProfiler features as it declines from 85% to 74%.

Conclusion

The predictions from graph features alone reach 43%, which is a medium to low precision. But this quality is actually surprising coming only from the extracted features from a graph with the position of the cell centroids and their connections. With RF or NN voting, adding graph features does not improve the predictions (RF) and in the worst case scenario impairs them (NN). It looks like the features extracted from the cell graph, mostly based on nodes and degree properties are not useful to differentiate treatments. The stronger degradation of results when adding graph features to the subpopulation proportions compared to when adding graph features to the CellProfiler features may be caused by the higher number of CellProfiler features: indeed the graph features bring then some noise to the data but they are not numerous enough to mess the decision boundaries. This effect is mainly visible with NN voting as the used cosine distance sets the same weight to every feature. On the other hand, random forest reduces the bad impact as it is able to put weights on important features and lowers the importance of useless ones.

I list here some ideas that could explain the bad performance of graph-extracted features:

- features extracted from the graph with *networkX* are not adapted to planar graphs, because they are mostly based on node degree. Nodes in planar graphs have a bounded degree: a cell cannot have much more than 6 to 8 neighbors.
- features extracted from the graph are not taking into account the subpopulation membership of the nodes or their neighbor relationships. The features are computed on a graph without node labels.
- the used normalization creates biases in the graph feature distributions, as DMSO images have a quasi continuous layer of cells, which is not the case for most drugs where cells are more sparse.

To overcome these problems, the idea is to compare directly the neighboring connections between subpopulations, as the t-test and the assortativity statistics showed that the co-localisation of subpopulations is not random. This is the subject of the next section about graph kernels.

4.5.2 With graph kernels

Comparison of 4 graph kernels

Methodology The idea of graph kernels is to compare directly two images without extracting features (cf Section 2.3.4). I tested 4 different types

of graph kernels from the GraphKernels *Python* library written by the Borgdwardt lab [193]: k-step random walk, connected graphlet, shortest path, Weisfeiler-Lehman (WL).

In a few words, k-step random walk creates lists of length k of the subpopulation labels of visited nodes on random walks, that are later compared. Connected graphlet counts the occurrences of all possible connected graphlets of k nodes, where k is usually given a small value. Connected graphlets in this implementation does not take into account the subpopulation node labels. Shortest path kernel computes the shortest paths between all pairs of nodes. Weisfeiler-Lehman kernel looks at how many nodes are in two graphs that have the same label and the same neighboring labels (cf Figure 2.9 from the Chapter State of the art). For each pair of graphs, the value of the graph kernel is computed, and put together in a large square matrix K.

I used two ways for MOA prediction:

- with a distance matrix D obtained from the kernel matrix K:

  $$D(G_i, G_j)^2 = ||G_i - G_j||^2$$
  $$= K(G_i - G_j, G_i - G_j)$$
  $$= K(G_i, G_i) + K(G_j, G_j) - 2K(G_i, G_j)$$

  Then I used the nearest neighbors (NN) voting with this new distance matrix. I tried a variable number of neighbors using majority voting.

- with a kernel Support Vector Machine (kSVM), with one SVM per drug, i.e. one vs all. The kernel matrix K is directly fed to the kSVM algorithm. The kernel space is used with the assumption that the separation between classes will be made easier than in the original space.

As previously, I used a leave-one-drug-out cross validation (LOO-CV) approach. I tested the prediction for each image and for each treatment for both distance-based NN and kSVM approaches. To obtain a prediction per treatment, I averaged the kernel values for all pairs of images coming from the same treatments, i.e. if $\{A_i\}_i$ are images from treatment A and $\{B_j\}_j$ are images from treatment B, then all the kernel values $\{K(A_i, B_j)\}_{i,j}$ will be averaged as the kernel value for the pair of treatments (A, B). It seemed the closest approach to the one previously used with subpopulation proportions. Then on this new smaller kernel matrix, I applied the prediction methods.

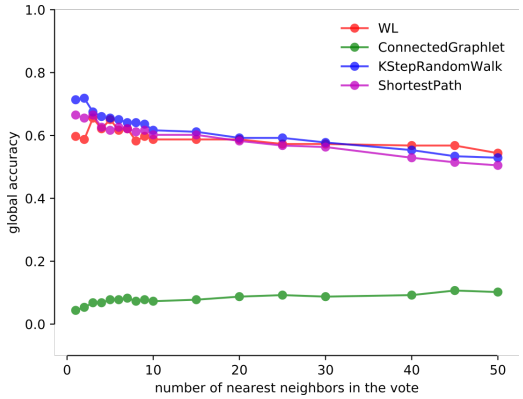


Figure 4.24: **MOA classification accuracies for 4 graph kernels and nearest neighbors voting per image.** To predict the MOA of one image, its k closest images are selected according to the graph kernel-derived distances. The MOA that appears in the majority of the closest images is selected as the one predicted. The 4 graph kernels are Weisfeiler-Lehman (WL), connected graphlet, k-step random walk and shortest path. These results were obtained with the GraphKernels *Python* library [193].

Results per image As seen on Figure 4.24, the MOA predictions per image are quite comparable for the 3 graph kernels WL, k-step random walk, shortest path and across different numbers of voting neighbors. However, connected graphlets kernel is making poor predictions. It is the only one of the four kernels that does not take the subpopulation labels into account. It makes sense that the planar graph structure alone cannot differentiate between MOAs, as

	kSVM p	Accuracy kSVM	Accuracy 1NN
K-step random walk	10	66	**69**
Shortest Path	10	67	67
WL	10	**68**	57
Connected graphlet	1	2	16

Table 4.2: **MOA classification accuracies for 4 graph kernel methods with kernel-SVM and nearest neighbor voting per image.** Logarithmically separated values between 0.001 and 100 were tested for the SVM parameter "p" and the one giving the best results is reported.

seen in the previous section with the extracted graph features (cf Section 4.5.1).

The kSVM results per image are reported in Table 4.2. We see the same trend as for the NN voting procedure. The 3 graph kernels WL, k-step random walk and shortest path give good accuracies averaged per image, while connected graphlet's accuracy is poor.

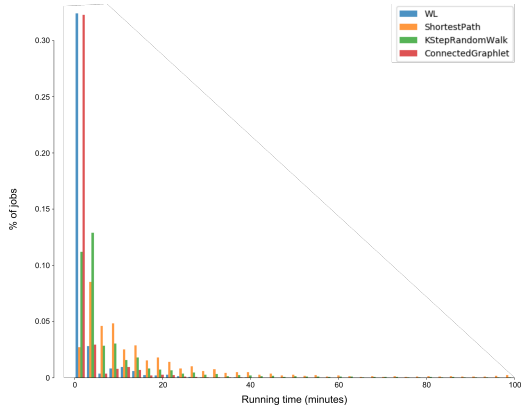


Figure 4.25: **Running times of the 4 different graph kernels.** One job corresponds to one pair of graphs. The GraphKernels *Python* implementation was used with one CPU and up to 100 Gb of RAM.

Results per treatment For the predictions per treatment, the global accuracies are reported in Table 4.3. For both the NN voting system and the kSVM, k-step random walk is doing slightly better than WL and shortest path, with 77% and 78% accuracy for k-step random walk compared to 70% and 71% for shortest path, and 67% and 45% for WL respectively. The NN voting system does not perform well with WL compared to the kSVM, whereas it is the opposite for connected graphlets.

Running time for the graph kernel jobs The implementation of the graph kernels in the GraphKernels *Python* library [193] is demanding in terms of resources. The running time of graph kernel jobs are displayed on the Figure 4.25. There is a high variability in terms of needed resources for different pairs of graphs. WL and connected graphlet are the less greedy in terms of time with the majority of jobs running under 4 minutes. However, for k-step random walk only 40% of the jobs finish before 20 minutes, and only 27% for shortest path. With more than 1,000 images, the number of image pairs is of the order of half a million. The running time of these jobs is then, even on a cluster, a crucial parameter. As the performances of the k-step random, shortest path and WL kernels are somehow comparable, and WL kernel much faster, I proceeded to further tests only with the WL kernel.

Impact of input parameter on WL graph kernel

Methodology For WL kernel, there is one parameter p that controls how many circles of neighbors are taken into account. It corresponds to the number of iterations done (cf Figure 2.9 page 48 from Chapter State of the art). It can also be considered as a scale parameter: a low value of p will take into account only the immediate neighbors, whereas a larger value of p will aggregate a larger

	kSVM p	Accuracy kSVM	Accuracy 1NN
K-step random walk	1	**78**	**77**
Shortest Path	1	70	71
WL	0.01	67	45
Connected graphlet	1000	11	26

Table 4.3: **MOA classification accuracies for 4 graph kernel methods with kernel-SVM and nearest neighbor voting per treatment.** Logarithmically separated values between 0.001 and 100 have been tested for the SVM penalty parameter p for the error term (cf scikit-learn SVC function [74]). The one giving the best results is reported. The kernel values are averaged per treatment before submitting to the kSVM (cf previous Methodology paragraph).

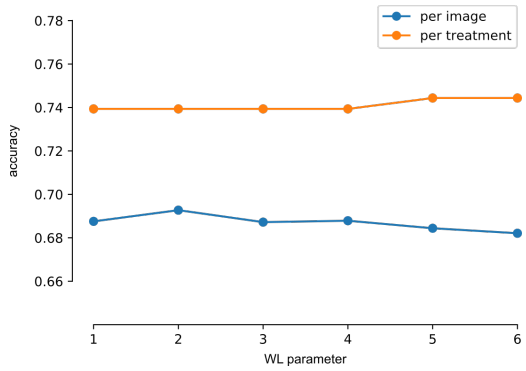


Figure 4.26: **Impact of the WL parameter on MOA classification accuracy per image and per treatment.** WL parameter $p = 1$ corresponds to no neighbor.

	per image	per treatment
kSVM	69	**76**
1NN	69	74

Table 4.4: **MOA classification accuracies for Weisfeiler-Lehman graph kernel method ($p = 1$) with kernel-SVM and 1NN voting per image and per treatment.** WL parameter is 1, so no neighbors are taken into account.

neighborhood. The case with no neighbor is $p = 1$, and when p increases, more and more neighbors can be compared. For the above results, the default value of the parameter was 5.

Results On the Figure 4.26, we can see the impact of increasing the neighborhood taken into account by the WL graph kernel, knowing that $p = 1$ corresponds to no neighbor taken into account. We see mostly no effect of the variation of the parameter. The WL graph kernel reaches a lower MOA classification accuracy, with maximum 74% when averaging the kernel values per treatment.

However taking only the proportions of the clustering subpopulations and using the Ljosa et al.'s LOO-CV with nearest neighbor voting reaches 85% accuracy. How to explain this difference? Let us take the case where $p = 1$, i.e. no neighbors are taken into account and compare the different results (cf Table 4.4). With Ljosa and collaborators' method, the subpopulation proportions are computed per image, then averaged per treatment, and only then the similarity is computed via a cosine distance. With graph kernels, the two steps of getting the image vector and the similarity computing (scalar product) are pulled into one, the graph kernel computation. So the numbers are not fully comparable.

On the other hand, we can distinguish effects of some variables (cf Table 4.4): kSVM method works better with the scalar product than the distance-based NN voting system for the per treatment prediction, and pooling results per treatment improves the accuracy (as already seen on Figure 4.26).

Impact of the number of clusters and the WL parameter on WL graph kernel

The previous result (cf Figure 4.26) was obtained with a PhenoGraph clustering giving 34 clusters. The rationale is that when pooling the subpopulation labels

with the ones from the neighboring cells, the combinatorial effect increases exponentially the number of possible labels for a node. If the number of possible labels is large, then the probability of having a sufficient number of comparable nodes between two graphs slims down. Hence I wanted to study the effect of the number of clusters on the efficiency of graph kernel methods.

Methodology 99 k-means clusterings for each value of parameter $k \in [2, 19]$ were computed. For each clustering, the graph kernel method WL was run with different parameters p from 1 to 6. The MOA classification accuracies were monitored via a kSVM both per image and per treatment.

Results On Figure 4.27, the MOA classification accuracies per image in relation with the WL parameter and the number of clusters in the k-means are displayed. We see that the more clusters there are, the better the accuracy is. For all values of k, going from $p = 1$ to $p = 2$ is marked by an increase of accuracy. This increase is more important for lower values of k. For all values of k, increasing the WL parameter to more than 2 does not increase the accuracy anymore.

When looking at Figure 4.28, i.e. results per treatment, the behavior is slightly different. The effect of the WL parameter is almost null, except for very low values of k such as 2 or 3. However when comparing the raw accuracy achieved by WL graph kernel to the one based on Ljosa and coworkers' procedure from the subpopulation proportions, WL graph kernel performs at the same level or higher. We find again that pooling per treatment allows a better overall performance.

Conclusion Aggregating neighbors is definitely helping average or low clusterings results with few categories. However as a lower value of k for k-means is associated to a lower MOA classification performance, with these results, it is hard to distinguish the effect of the number of clusters, k, and the initial level of performance before adding more neighbors. The greater increase for lower numbers of clusters can be due to the lower number of added labels when pooling the different rounds of neighbors, then the number of possible node labels is low enough to make fruitful comparisons.

For the WL graph kernel, the vector representing one graph is longer and longer as p increases: it keeps the original labels and concatenates the added labels (cf Figure 2.9 from Chapter State of the art). Hence if the number of original labels is relatively high, the representation of the added node labels is very sparse and it is very unlikely to have two images having some nodes hitting the same added labels. Hence the impact of the scalar product of adding new rounds of neighbors is probably insignificant.

4.6 Discussion

In this result section, I explored how to extract subpopulations from cell features, how these subpopulations were spatially arranged, and if this spatial organization could be leveraged for drug profiling. This work was entirely based on the BBBC021 dataset where 38 drugs were tested at different concentrations on MCF-7 breast cancer cells and recorded with three fluorophores targeting DNA, Actin and Tubulin.

CellProfiler was used to segment and extract many features for each of the seg-

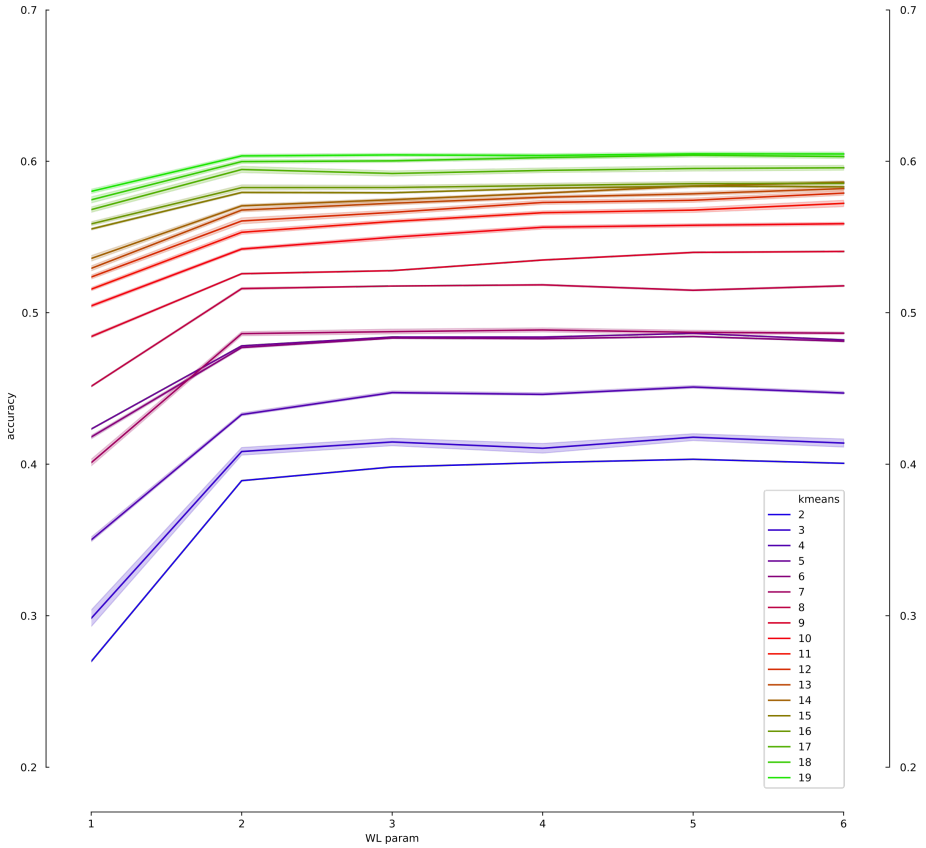


Figure 4.27: **MOA classification accuracies for k-means clustering and WL graph kernel method per image.** 99 runs of clusterings and graph kernels were computed. The mean MOA classification accuracy and their standard deviations are displayed in function of the different values of the WL graph kernel method parameter.

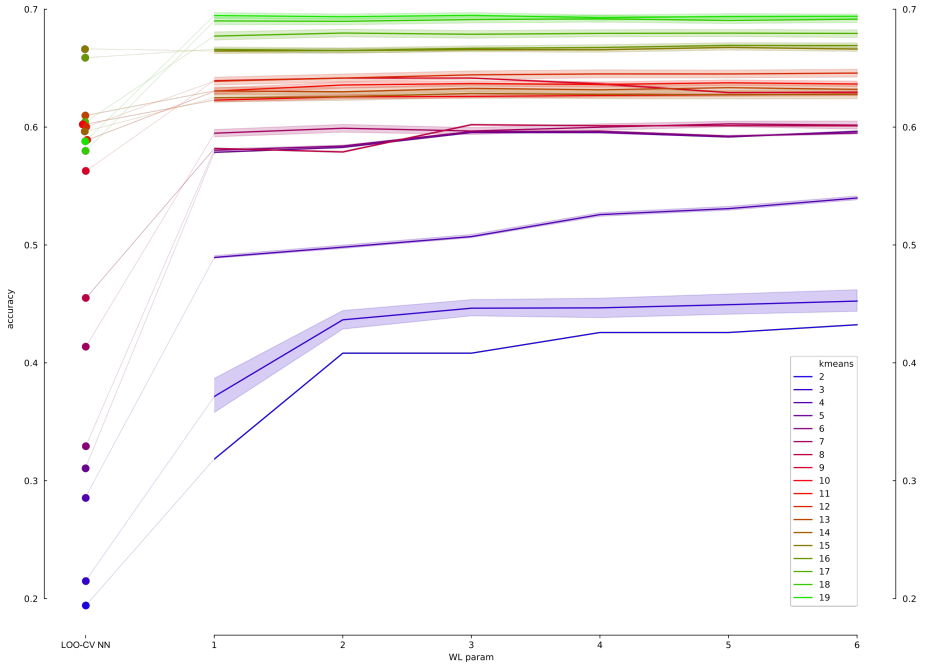


Figure 4.28: **MOA classification accuracies for k-means clustering and WL graph kernel method per treatment.** 99 runs of clusterings and graph kernels were computed. The mean MOA classification accuracies and their standard deviations are displayed in function of the different values of the WL graph kernel method parameter. These accuracy values can be compared with the one obtained with the leave-one-drug-out cross-validation (LOO-CV) nearest neighbor (NN) voting on the proportions of subpopulations.

mented cell. Some basic exploration of the CellProfiler set of features showed that they are highly correlated, which was already known [17, 25, 33]. However, the fact that features are correlated limits the bad effect of the curse of dimensionality, and makes the small distances in the feature space still reliable: closest neighbors are cells that have a relatively close phenotype from the query cell.

From these features, the goal was to use clustering methods to group similar cells across treatments and obtain cells that have the same phenotype in groups called subpopulations. I tested classical clustering methods like Gaussian Mixture Model (GMM), k-means or Support Vector Machine (SVM), and more ad hoc methods like flowSOM based on Self-Organizing Maps and PhenoGraph based on the Louvain's algorithm. With criteria like the mechanisms of action (MOA) accuracy and the replicate similarity, k-means, flowSOM and PhenoGraph achieves satisfying results overall. Phenograph takes an interesting approach as it does not require as input the number of clusters, but deduces it from the optimization of a quantity called modularity. The observation of the subpopulations is rather satisfying under eye inspection. The performances of PhenoGraph are also not too sensitive to the input parameters, that are the number of neighbors to construct the similarity graph and the mininimal cluster size. As far as the reproducibility of clustering is involved, k-means performs better than PhenoGraph and flowSOM which have about similar performances. However, most likely, the smart initialization of k-means available in the *Python* package scikit-learn may be the cause of this stability.

When comparing the MOA classification accuracies between the benchmarked clustering methods used here and the diverse unsupervised methods benchmarked by Ljosa and collaborators, we can see that they all hit the same range, around 83%, with the exception of factor analysis.

By a custom-developed tool based on probabilities and t-test, and by a mixing statistics, Newman's assortativity, I showed that subpopulations are not randomly organized on the image space, but are localized next to each other more than expected at random. This property appears true for most of the subpopulations. Features extracted naively from the cellular graph do not improve the quality of the feature set in order to proceed to the MOA classification, even though they are capable of some separation between MOAs by themselves. An approach by graph kernel methods which directly compares neighborhoods of cell subpopulation labels between pairs of images appears suitable to grasp the colocalization manifestation seen previously. Graph kernel methods reach satisfying accuracy but do not allow to surpass the simplest method, the averaging of CellProfiler features. When using the graph kernel methods on k-means clustering results, I showed that using the Weisfeiler-Lehman (WL) graph kernel improves the MOA accuracy especially for low values of k. It would be interesting to test this approach on a dataset where there would be a few detected subpopulations and a low basal MOA classification.

All along this exploration, I used the mechanisms of action (MOA) metadata to assess the biological meaning of any clustering or sets of features. There are different ways to predict the MOA, I used for the most part Ljosa et al.'s method of leaving one-drug-out and predicting the MOA based on the closest neighbor treatment. Averaging the features per treatment improves signifi-

cantly the prediction by smoothing the potential outliers. However, when computing features directly per image as for the cell graph features or the graph kernels, using the same procedure than for features per cell is questionable: they are less values to average, hence the law of large numbers does not apply and the smoothing may not be as effective. Also for graph kernels should the averaging be done after the kernel computation? Indeed, the kernel value is more comparable to a distance rather than to a raw feature. Yet in my work, I averaged directly the kernel values. In the case of WL graph kernel, some implementations like the one in *Matlab* [209] give access to the intermediate ϕ vector representing the number of each node label (cf Figure 2.9 page 48 from Section State of the art). This ϕ vector is itself more comparable to a vector of pooled features per treatment, and could be averaged before computing the scalar product, mimicking better the original procedure of Ljosa and coworkers. Moreover as we look at slight changes in the percentage of MOA classification accuracy, these details of procedure can impact the results.

Chapter 5

A Python package for spatial analysis of cell images as a toolbox to answer biological questions.

5.1 Introduction

This work was done in collaboration with the Alexandrov team, specialized in Spatial Metabolomics at the European Molecular Biology Laboratory (EMBL, Heidelberg, Germany) [210]. Their team focuses on developing experimental and computational tools to study the spatial organization of metabolic processes. They mainly develop and maintain MetaSpace, a web-based tool to help molecular annotation of large amount of data generated by high resolution mass spectrometers [211]. Their Matrix-Assisted Laser Desorption/Ionization (MALDI) instrument allows them to visualize hundreds of metabolites with spatial resolution down to 5 μm in both 2D and 3D.

In particular, they recently developed a single-cell mass spectrometry technique to study metabolites of 2D cell samples [61]. In a few words (cf Figure 5.1), they combine two microscopy images with one image from a MALDI mass spectrometer (MS). Indeed, they proceed in 3 imaging steps. First, they image with a regular optical microscope the cells, with or without fluorescent markers. Secondly, they image with the MALDI MS according to a regular 2D grid. The laser, used to ionize the cellular molecules, deteriorates the coated matrix and leaves ablation marks that can be seen under an optical microscope, that corresponds to the third imaging step. They designed a data integration pipeline to extract and match relevant single-cell information. The data integration pipeline is composed of a registration step, a normalization step, and a metabolite and cell selection step. The registration step allows to superimpose the three images, namely the pre-MALDI microscopy image, the MALDI image and the post-MALDI microscopy image. This way, the measure points from the MALDI MS can be localized in the microscopy image space. Then, the normalization step assigns to each cell the portion of metabolites contained in each measure point, in proportion of the area intersection between the cell and the ablation mark. Finally, they filter the metabolites and only select the ones which pixel values are correlated with the presence of cells. Indeed, some of the detected molecules belong to the matrix or the media and not to the cells themselves, and would not be relevant to the following analyses. To summarize, this method produces single-cell levels of hundreds of metabolites along with a microscopy image, where other cell features like size, shape, fluorescence intensity, neighborhood can be assessed.

They tested their pipeline on different biological examples. One of them is the study case of hepatocyte-like cells, investigated by the team of Mathias Heikenwälder at the Deutsches Krebsforschungszentrum (DKFZ, German Cancer Research Center, Heidelberg) [212]. Inflammation can be induced in this cell line, and monitored through a fluorescent marker (details in the following section). Cell inflammation for hepatocytes translates into a cytoplasmic accumulation of lipids [213, 214]. The single-cell MALDI technique developed by the Spatial metabolomics team is able to pinpoint the nature of the accumulated lipids. The results of this investigation are reported in Rappez and collaborators [61].

Another biological example they studied is the co-culture of Hela cells and fibroblasts NIH3T3. This co-culture with very different cell types allows to test the spatial resolution of the whole pipeline. Indeed, with a subset of chosen metabolites they can predict the cell type with a 96.6% accuracy. It shows that there

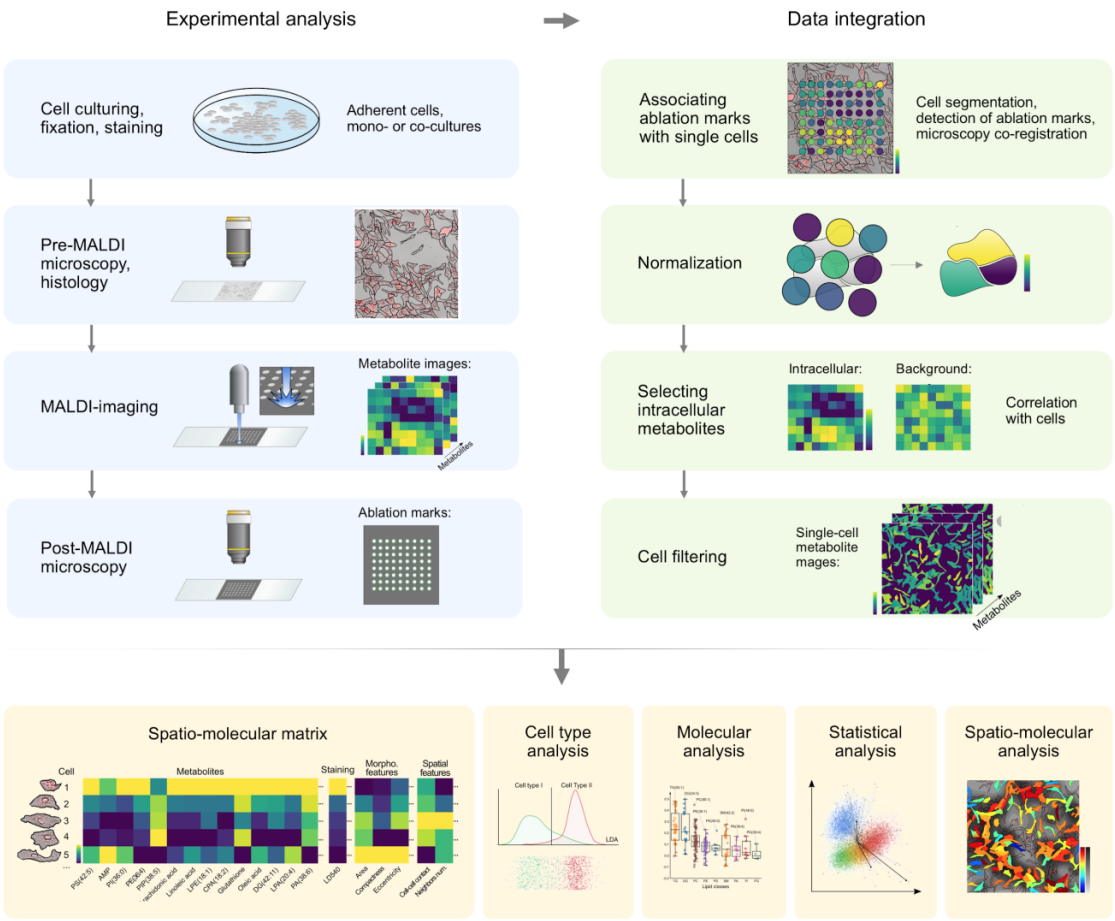


Figure 5.1: **Method for spatial single-cell metabolomics by integrative microscopy and MALDI imaging mass spectrometry.** First column: Experimental analysis. The cells are first cultured in monolayer, fixed and stained. Then a first microscopy image is taken, before the MALDI imaging. The post-MALDI image is needed to visualize the ablation marks made by the mass spectrometry laser. Second column: Data integration. Registration between the three images is possible thanks to the ablation marks. Once the cells segmented from the pre-MALDI microscopy image, each ablation marks is associated to one or more cells. Depending on the intersection between the area of the ablation mark and the area of the cell, a proportion of the related metabolites measures is associated to each cell. By looking at the correlation between metabolites and presence of cells, some metabolites not correlating with the presence of cells are discarded, the extra-cellular metabolites. Hence a data matrix is constructed with a value for each segmented cell and each intracellular metabolite, leading way to possible cell type, molecular, statistical and spatio-molecular analyses.

is little to no spatial leaking of metabolites from one cell to the other. They are also currently working on a timelapse dataset, where they follow cells undergoing divisions and migration with live microscopy and at the end of the experiment, the cells are fixed and MALDI-imaged.

The collection of these spatial data drives new questions. They noticed that for the hepatocyte data, highly fluorescent cells seemed to be grouped in small patches, and be positioned around other cells. Is this observation statistically significant? Does it display a robust association? For the co-culture experiment, they asked whether the cell populations are lying randomly. However, the two cell populations are not in even proportions and it biases the human perception of spatial randomness or the raw counts of nearest neighbor cells. For the time lapse experiment, we can ask the question of whether the cells share more similar metabolites when they originate from the cell division or if they share a similar micro-environment. Unfortunately, this was an ongoing experiment at the time of my visit and I did not have access to good enough segmented and tracked data to start developing spatial analysis.

I wrote *PySpacell*, a *Python* package dedicated to cell spatial image analysis, to answer some of the questions we came up with during this collaboration. We described and illustrated it in the following article. The main features of the toolbox is its easy installation via pip, its compatibility with image analysis software outputs, its versatility for cell neighborhood definition and for the type of possible cell features that can be tested.

5.2 PySpacell : A Python package for spatial analysis of cell images

PySpacell : A Python package for spatial analysis of cell images

France Rose[1], Luca Rappez[2], Sergio H. Triana[2], Mira Stadler[3], Mathias Heikenwalder[3], Theodore Alexandrov[2], and Auguste Genovesio[1]

[1] Computational Bioimaging and Bioinformatics, Institut de biologie de l'Ecole normale supérieure (IBENS), Ecole normale supérieure, CNRS, INSERM, PSL Université Paris 75005 Paris, France.

[2] Spatial metabolomics. European Molecular Biology Laboratory, Heidelberg, Germany.

[3] Division of Chronic Inflammation and Cancer, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 242, 69120 Heidelberg, Germany

Abstract

Technologies such as microscopy, sequential hybridization, mass spectrometry and others enable to collect quantitative single cell read-out measurements in situ. However, spatial information is usually overlooked by downstream data analyses, which usually consider single cell read-out values as independent measurements for further averaging or clustering, thus disregarding spatial locations. With this work, we attempt to fill this gap. We developed a toolbox which enables to easily compute spatial statistics from cell images in order to test if and at what scale features of interest display a non random spatial trend. The proposed Python module encompasses regular cell image data as well as other type of single cell data such as in situ transcriptomics or metabolomics. Input format of our package matches standard output formats from image analysis tools such as CellProfiler, Fiji or Icy, and thus make our toolbox easy and straightforward to use, yet offering a powerful statistical approach for a wide range of applications.

Availability: Python 3.5 package. Pip install. Github. Jupyter notebook tutorials.

Keywords: spatial analysis, single cell, microscopy, statistics, imaging, spatial dependence, spatial heterogeneity, spatial autocorrelation, neighborhood matrix, cell graph

Introduction

Single cell measurements is now at Biologists' reach. With new technologies such as single-cell RNA sequencing (Zhu et al. 2018; Lubeck et al. 2012; Wang et al. 2018), microfluidics developments, matrix-assisted laser desorption/ionization mass spectrometry (MALDI MS; Chen et al. 2016), one can monitor cell-to-cell differences which would otherwise be hidden in population measurements. These cell-to-cell differences are due to many factors such as the stochasticity of gene expression, and the variety of proteomes and metabolomes (Chen et al. 2016). Cell-to-cell variation is also the result of the cell microenvironment, and therefore it can be related to the states of its neighboring cells. Especially, cell spatial heterogeneity may reflect communication between cells to control position-specific cell states (Zhu et al. 2018). Snijder and collaborators (Snijder et al. 2009) used cell positional information to predict single cell viral infection levels. They showed that, even in a cultured cell context, cell spatial microenvironment influences the cell state. Additionally, two teams (Toth et al. 2018, Rohban et al. 2018) recently found that considering single cell features computed using the states of neighboring cells improved drug profiling and cell phenotype classification in high content screens (HCS). Thus, neighboring cell states seem to bring additional information, reinforcing the need to study and visualize spatial heterogeneity in microscopy images.

Since the recent developments of spatially aware experimental techniques to study gene expression (Stahl et al. 2016; Wang et al. 2018), the interest for spatial analysis has increased. Indeed in the last year, computational methods to analyse spatial gene expression data were developed independently by several teams (Svensson et al. 2018; Edsgärd et al. 2018; Keren et al. 2018). Using different methodologies, such as Gaussian process regression (Svensson et al. 2018) or summary statistics on marked point processes (Edsgärd et al. 2018), these approaches intend to identify genes whose expression is spatially variant, to group them into similar spatial patterns (Svensson et al. 2018), and to find local hot spots (Edsgärd et al. 2018). However, these tools are not easily applicable to the analysis of cell images, as they are designed to process gene read counts in order to identify a small subset of genes whose expression varies spatially. In this context we found out a lack of a versatile toolbox that could detect, quantify and compare spatial patterns of single cell features.

With this work, we propose a tool that takes as input preprocessed cell images obtained for instance by fluorescence microscopy, and outputs several quantities providing information about the spatial non-randomness of any given feature spanning across the whole image. For instance, features can be the level of expression of a gene, the fluorescence intensity of a reporter, the quantity of a metabolite, the cell type, the location of a fluorescently tagged protein, etc. The read-out of interest can be real-valued, e.g. the quantity of a protein, or categorical, e.g. a protein located in the cytoplasm or in the nucleus. These spatial non-randomness measures can be compared between samples or between different scales on the same sample. To our knowledge, it is the first tool dedicated to spatial statistical analysis of cell image data.

Methods

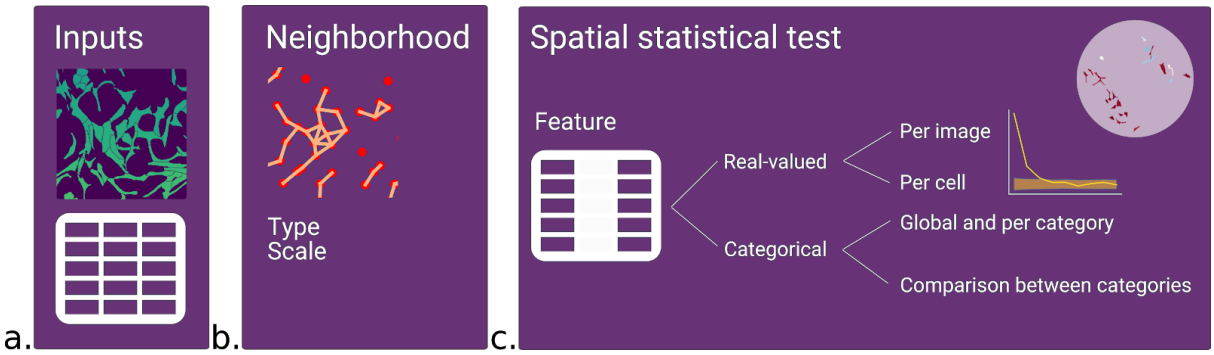


Figure 1 - Overview of PySpacell pipeline and use cases. a. PySpacell takes as inputs a preprocessed image with pixel values corresponding to unique cell ids, and a tabulated file containing the following columns: id, x and y coordinates of the cells, and as many additional columns as there are measured features. **b.** Then a neighborhood type and scale need to be chosen according to the data (distances are in pixels). **c.** Finally, a feature is selected along with a compatible spatial test. If the feature is quantitative, spatial tests can be run per image or for every cell object of the image. If the feature is categorical, spatial tests can be run per image, separating the categories or pulling all the categories together. Several tests can be run and stored for further comparisons and visualizations.

General purpose

PySpacell provides spatial statistics for cell images. It implements and adapts methods developed in network analysis (Newman 2003) and ecology fields (Dixon 2002, Legendre and Legendre chap 13 1998) where data are modeled as marked point processes.

Input

PySpacell only requires a label image, where each pixel that belongs to a cell has a value matching a unique cell id, and a csv file containing single cell feature information for each of these cell ids (Figure 1a.).

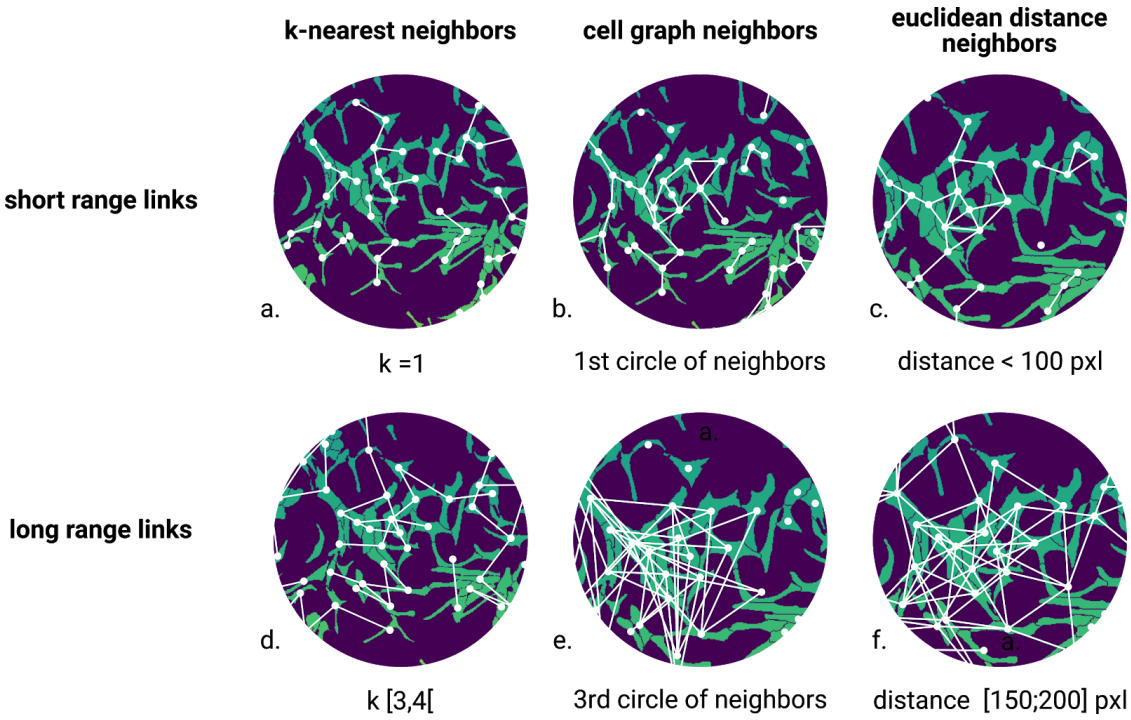


Figure 2 - Selection of a neighborhood definition. The feature value of a given cell will be compared in spatial statistical tests to the feature values of the chosen neighbors. Three neighborhood types are available: k-nearest neighbors (a. & d.), cell graph neighbors (b. & e.), euclidean distance neighbors (c. & f.). Additionally a distance range allows to filter out cells laying close by or far away.

Neighborhood types

In PySpacell, the user can choose among several definitions of cell neighborhood and test each of the features of interest on them to see if there is a non-random spatial pattern at certain scales. The tool includes three neighborhood definitions: k-nearest neighbors (Figure 2a. & d.), cell graph neighbors (Figure 2b. & e.) and euclidean distance neighbors (Figure

2c. & f.). For k-nearest and euclidean distance neighbors computation, euclidean distances between all pairs of cells are computed. For network neighbors computation, the cell graph is computed from the label image: cells are neighbors if their membranes are in contact. An `iterations` parameter can be used to relax the membrane touching constraint: in that case two cells can still be considered as neighbors if the shortest distance between them across background pixels is less than a selected value. The choice of the neighborhood type is tightly related to the aim of the study. For instance, if the observed process is likely to act through a diffusion mechanism, then 'radius' seems to be a good choice, however if it is likely to act through cell-to-cell contacts, then in such case 'network' seems more reasonable. Furthermore, choosing a scale range, by setting a minimum and a maximum parameters, allows to target short (Figure 2a., b.& c.) or long (Figure 2d., e. & f.) range relationships. For example for euclidean distance neighbors computation, all the cells situated within a certain radius range will be kept. This way, a range of scales can be investigated and compared. Once selected and computed, the neighborhood can be visualized with the `plot_neighborhood` function (Figure 2).

Computed spatial statistics

The study of spatial patterns for quantitative features is achieved through computation of spatial autocorrelation indices, which compare the feature value of one cell to feature values of its neighboring cells, and check if such a spatial clustering of values is statistically significant or possibly drawn from a null model (Legendre and Legendre 1998). Feature's spatial organization can be computed for the whole image (one statistical test per image), or for each individual cell (one statistical test per cell). Several spatial indices are available: Moran, Geary and Getis-Ord for global statistics, and Moran and Getis-Ord for local statistics (Dale & Fortin 2005). This part of the calculation is undertaken by the Python module *pysal*

(Rey et al. 2010). The significance of each statistics is evaluated with random permutations of single cell feature values. The number of permutations can be set by the user.

Two types of spatial analysis are provided for categorical features: Newman's assortativity (Newman 2003) and Ripley's K cross functions (Dixon 2002). The significance is also tested with category permutations. The number of permutations is up to the user. The general principle of Newman's assortativity analysis is to count existing links between cells of different categories, and to compare these counts to the expected number of links under the null model. The general principle of Ripley's K cross-functions is to count how many objects are within a distance from a given object, then average and normalize this value over the dataset. The Ripley's K cross-function K(i,j) counts the number of objects of category j around objects of category i. Computing the difference K(i,j) - K(i,k) allows to measure if there are more objects of category j or k around objects of category i. With these cross functions, every possible object category pairings can be investigated. The assortativity and Ripley's functions approaches are complementary. On one hand, the all-class assortativity provides an answer to the question "are cells of the same class more frequently linked to each other than expected at random?", and the assortativity per-class i provides an answer to the question "are cells of class i more frequently linked to each other than expected at random?". On the other hand, Ripley's K function provides an answer to the question "are cells lying randomly assuming a random Poisson process?", and Ripley's K cross function difference K(i,j) - K(i,k) measures whether instances of category j are more grouped around instances of category i than instances of category k. Ripley's computation output can be variance stabilized (function L) or not (function K) depending on an optional parameter.

Test results are stored in a *pandas* (https://pandas.pydata.org/) dataframe named `perimage_results_table` if it is a global test, or directly in the input pandas dataframe named `feature_table` otherwise.

Visualization

To visualize a test computed using one feature from a single image at different scales, the chosen statistics can be easily plot against increasing scales with the `plot_correlogram` function. It works for test results on both categorical and quantitative features. A correlogram can help to estimate if there is positive or negative spatial similarity, and at which scale (Legendre and Legendre 1998). The scales displayed on the x-axis should be interpreted in two different ways depending on the chosen statistic. On one hand, the theoretical model of the Ripley's functions considers a growing neighborhood that goes up to a certain radius (Dixon 2002, Figure 3d.&e.), meaning that the tests at each radius r are computed with the neighborhood parameters ('radius', 0, r). On the other hand, for spatial autocorrelation indices, we would advise to use successive ring-shaped neighborhoods corresponding to different distance ranges (Legendre and Legendre 1998, Figure 2b.&d.), e.g. the test at each radius r_i is computed with neighborhood parameters (neighborhood type, r_{i-1}, r_i). These distance ranges allow to separate the effect of close and distant neighbors.
Per cell results can be visualized on the image space with the `get_feature_filled_image` and `plot_2im` functions.

The analysis and plot generation (Figure 3) for the hepatocytes image took about 5 minutes and about 1.7G on a regular laptop. The image size is 5154x5136 pixels and it contains 3256 detected cells. The analysis and plot generation (Figure 4) for the co-culture image took about 19 minutes and about 0.5G on a regular laptop. The image size is 3772x3718 pixels and it contains 2122 detected cells.

Results

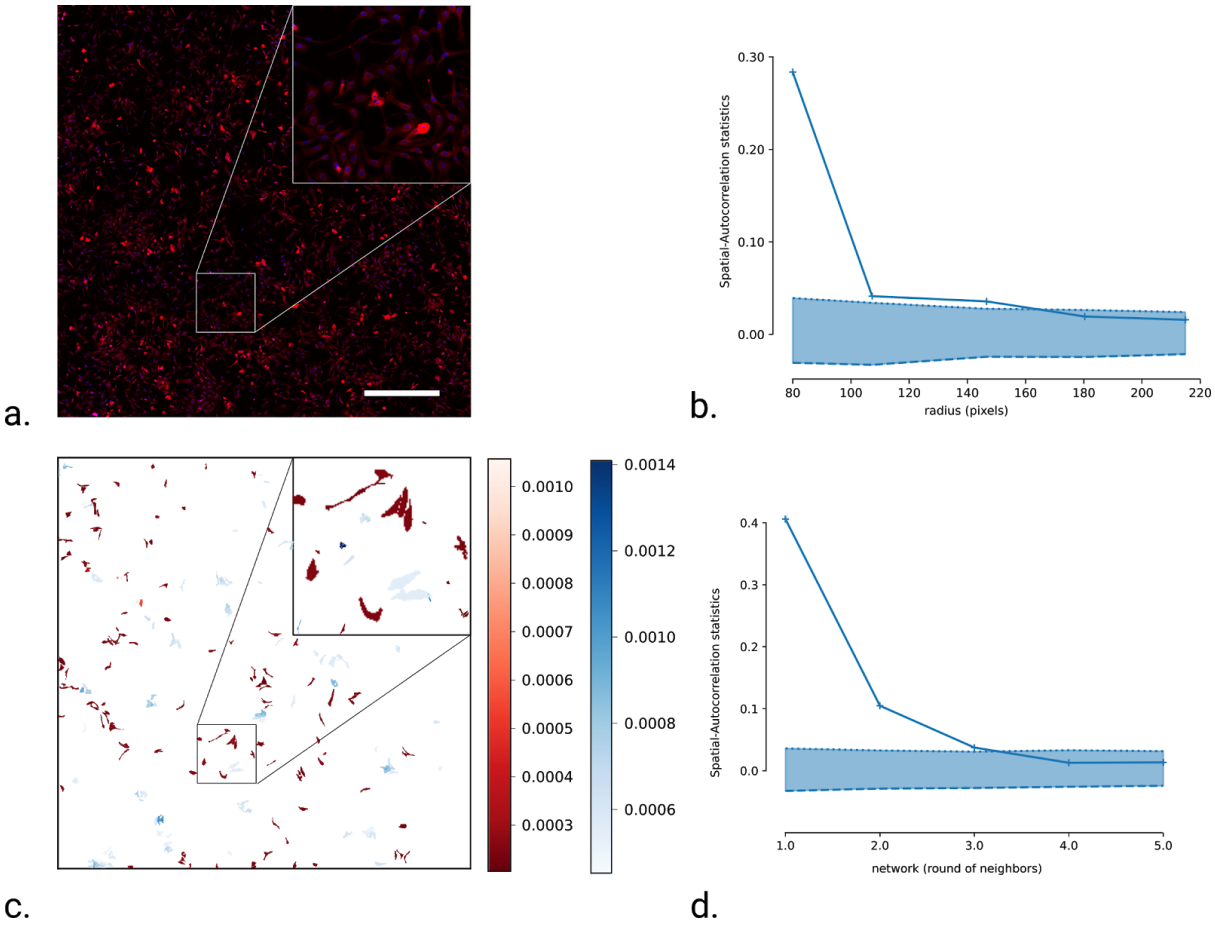


Figure 3 - Spatial clustering analysis of inflammation response from hepatocyte-like cultured cells a. Merged microscopy image of brightfield, DAPI (blue) and lipid droplets fluorescence (red) channels. The red intensity monitors inflammation of the hepatocytes via a lipid droplets marker, LD540 (Rappez et al. 2019). Scale bar: 700 µm **b.** Correlogram of Moran's statistics for the lipid droplets' mean fluorescence intensity per cell. The value at radius r_i is the Moran's statistics for all pairs of cells at a distance comprised between r_{i-1} (the previous x axis point) and r_i. The 2.5 and 97.5 percentiles of the null model were computed with 999 random permutations of the fluorescence values over the cells. **c.** Cells with Getis-Ord statistics above the 97.5% percentile (cold spots) are displayed in blue and those under the 2.5% percentile of the null distribution (hot spots) are displayed in red. Colormap represents values of the local Getis-Ord statistics. Cell neighborhood computed with parameters ('radius', 0, 79.983). **d.** Correlogram of Moran's statistics for the lipid

droplets' mean fluorescence intensity per cell using a network neighborhood. X-axis values for the network neighborhood correspond to circles of neighbors: x=1 corresponds to the membrane touching neighbors; x=2 corresponds to the neighbors of the membrane touching neighbors exclusively; and so on. A radius matching each circle of neighbors was calculated (see **b.**) to obtain about the same number of pairs in the two neighborhood matrices. The network neighborhood produces a higher and wider spatial autocorrelation for the lipid droplets fluorescence than the radius neighborhood.

PySpacell can detect spatial patterns displayed by a fluorescent single cell read-out

Hepatocyte-like cells (hepaRG) were induced by TNFα in combination with oleic and palmitic acids to produce an inflammation reaction. This inflammation reaction was monitored by the fluorescent marker LD540 (Figure 3 a., Rappez et al. 2019), that stains the cytoplasmic lipid droplets. The microscopy image displayed on Figure 3a. was segmented by Rappez and collaborators using CellProfiler (Rappez et al. 2019). A biological question that arises naturally from this experimental model is: is the inflammation state of one cell influenced by the inflammation states of the surrounding cells? To answer this question, PySpacell can compute the Moran's statistics on the mean LD540 intensity per cell, as a proxy for the inflammation state. The Moran's statistics is an autocorrelation measure that PySpacell can compute at several scales: for instance, the data point at radius 180.344 pixels on Figure 3b was obtained by setting respectively the minimum and maximum radius parameter at 146.601 and 180.344 pixels. The correlogram on Figure 3b. shows that Moran's statistics values for the mean fluorescence are positive, which means that similar values spatially cluster together. Furthermore, the computed spatial autocorrelation values are above the 97.5% null distribution percentile for radii smaller than 180.344 pixels: it means that under a 180.344 pixel distance, pairs of cells show a statistically significant clustering of similar

intensity values. We observe a spatial clustering of similar inflammation levels at small scale, as read from the LD540 fluorescence mean intensity per cell.

The choice of neighborhood definition provides a clue on the cell communication mode

To test if the spatial autocorrelation of hepatocytes' inflammation is likely due to a diffusion process or to a membrane-mediated cell-to-cell interaction, it is possible to compare the values of the same statistics obtained using two different neighborhood definitions. Indeed there are two natural ways to define the neighborhood of a cell: either it comprises all the cells within a given distance around it, or all the cells that shares a membrane with it. In the first case only pairs of cells at a certain distance are considered, and in the second case, the neighborhood is growing by aggregation of touching neighbors. By comparing the correlograms (Figure 3b. & d.) based on the two different neighborhood definitions, one can see the impact of the neighborhood choice on spatial autocorrelation. In Figure 3d., the spatial autocorrelation statistics has a value of 0.42 for the first circle of neighbors, which is higher than the value of 0.28 obtained in Figure 3b. for the first tested radius. The plots 3b. and 3d. were constructed such that their abscisses match in terms of cell pairs' count, preserving the statistical power of the spatial autocorrelation tests. Interestingly, in Figure 3d. the second data point lay way above the 95 % confidence interval of the null distribution (see Methods section), while in figure 3b. the second data point remains only slightly above. This comparison indicates that considering a neighborhood based on cell membrane touching enables to measure a larger spatial spread of the inflammation values than with a neighborhood based on euclidean distance. The space being considered homogeneous, such a statistical clustering implies an interaction between cells. Here, positive spatial autocorrelation results suggest this interaction is direct and happens through cell-to-cell membrane contacts.

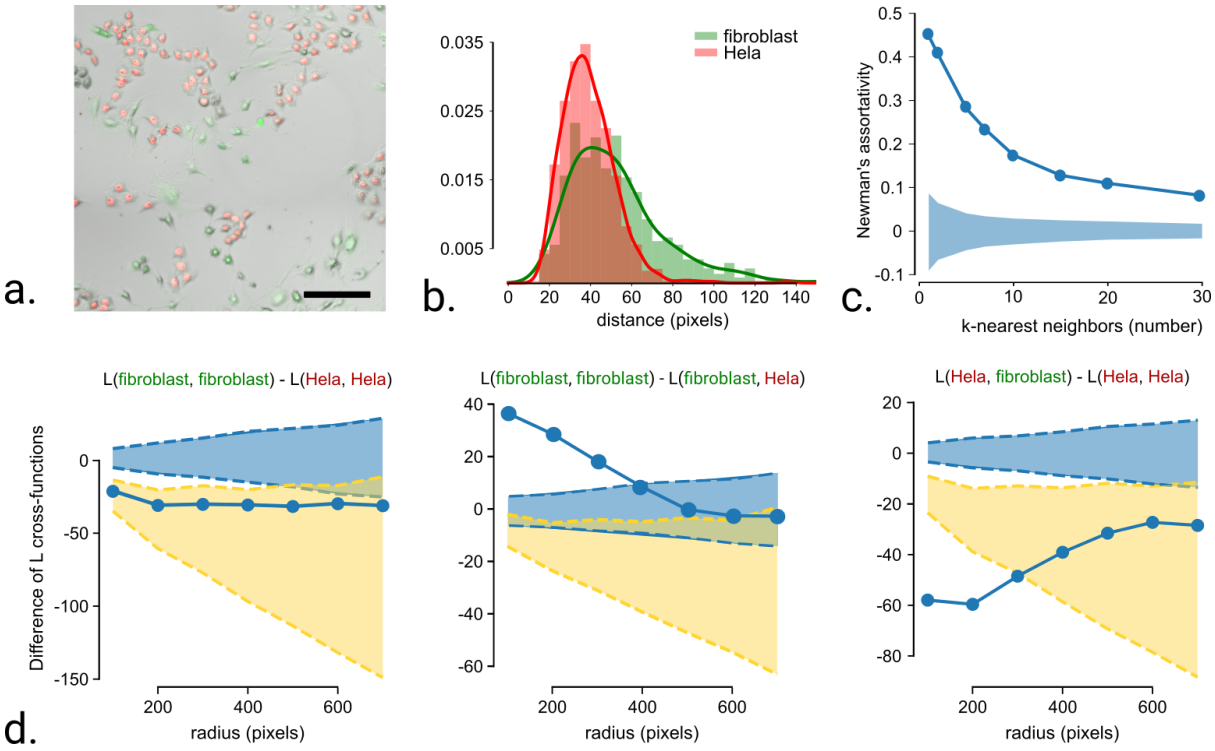


Figure 4 - Spatial clustering analysis of co-cultured cells. a. Crop of a merged microscopy image of phase contrast with a fluorescent Hela cells (in red - expressing H2B-mCherry) and fluorescent NIH3T3 fibroblasts (in green - expressing GFP) co-culture. Scale bar: 200 µm **b.** Histograms of the nearest neighbor distance for Hela cells and fibroblasts populations. Smooth curves are the Gaussian kernel density estimation of these histograms (bandwidth=0.22). **c.** Cell type assortativity computed with k-nearest neighborhood. The shaded blue area corresponds to the [2.5%, 97.5%] percentiles of the null distribution. The null distribution was obtained by random shuffling of the cell types. **d.** Plots of Ripley's L cross functions differences describing the relative spatial clustering of Hela cells and fibroblasts. The shaded blue area corresponds to the [2.5%, 97.5%] percentiles of the regular null distribution. The regular null distribution was obtained by random shuffling of the cell types. The yellow shaded zone corresponds to the [2.5%, 97.5%] quantiles of the constrained null distribution. The constrained null distribution was obtained

by randomizing cell types according to the nearest neighbor distance distributions (cf **b.** and see text for details).

PySpacell can detect spatial pattern displayed by cell categories

Hela cells and NIH3T3 fibroblasts (Figure 4a., Rappez et al. 2019) were co-cultured and seeded three days on a glass slide before imaging. HeLa and NIH3T3 cells constitutively expressed H2B-mCherry and GFP respectively. Cells were detected manually with Cell Counter ImageJ plugin [https://imagej.net/Cell_Counter, ImageJ]. An assortativity analysis (Newman 2003) was performed to count existing links between Hela cells and fibroblasts, and to compare this count to the expected number of links under the null model (Figure 4c.), based on k-nearest neighbors. For all k values, the data assortativity is greater than the null model (see Methods section), meaning that there is a higher number of links between cells of the same type than expected randomly. The null model was obtained by shuffling the cell type labels. A second tool was employed to compare the spatial clustering of one class to the other: Ripley's cross functions (Dixon 2002) allow to compare grouping, class per class. Indeed it can compare how many Hela cells versus fibroblasts lay in average around a Hela cell, by computing the difference of the normalized averaged counts (see Methods section). The null model confidence interval is also computed by label shuffling (Figure 4d. Blue shaded area). Results displayed on the middle panel of Figure 4d. show that fibroblasts are more likely to be surrounded by fibroblasts than Hela cells at small scales (radius < 500 px) as indicated by the positive difference. However, Hela cells are more likely to be surrounded by Hela cells than fibroblasts at all tested scales. Hela cells are also more likely to cluster than fibroblasts at all tested scales, as indicated by the negative difference (right panel of Figure 4d.). These values lay outside the null distribution confidence interval except for L(fibroblast, fibroblast) - L(fibroblast, Hela) above a 400 pixel radius (see blue shaded area

on the middle panel of Figure 4d.). It means that above a 400 pixel radius, this observation can be explained by the null model.

PySpacell limits the bias introduced by the size difference between cell categories

To lower down the effect of a possible bias introduced by the differences in size between the cell types, we propose, as an option, the construction of a null distribution through label shuffling that takes into account the cell type size (see yellow shaded areas on Figure 4d.). The area required by a cell is then approximated by the distance to its nearest neighbor. The histogram of this distance is plotted separately for Hela cells and fibroblasts on Figure 4b., Hela cells display a smaller nearest neighbor distance in average than fibroblasts. In order to respect the relative spatial occupation of both cell types, a new label is assigned to each cell depending on a probability computed from the nearest neighbor distance distributions' difference (Figure 4b.). The null model confidence intervals obtained this way are displayed in yellow on Figure 4d. L(fibroblasts, fibroblasts)-L(fibroblasts, Hela) is positive up to a radius of 500 pixels, but remains outside the 97.5% null model confidence interval when label shuffling is constrained by the size distribution difference (see middle panel of Figure 4d.). This demonstrates that the size difference taken alone cannot explain the fact that fibroblasts are more frequently surrounded by fibroblasts than by Hela cells. However, L(Hela, fibroblasts)-L(Hela, Hela) is negative, and outside the confidence interval up to a radius of 400 pixels only, when shuffling is constrained by the size distribution difference (see right panel of Figure 4d.). It means that the clustering of Hela cells around Hela cells can be explained by the difference in size beyond a radius of 400 pixels. Finally, L(fibroblasts, fibroblasts)-L(Hela, Hela) is negative, and inside the null model confidence interval for all tested radii (see left panel of Figure 4d.). It means that the size dependent model can explain that Hela cells appear to cluster more than fibroblasts at all tested scales.

Discussion

PySpacell is a toolbox for spatial statistics on cell images that enables to easily test if interesting cell features show a spatial trend and to what extent.

We demonstrated the capabilities of PySpacell on two fluorescence microscopy image examples. First we showed that hepatocytes display spatially clustered inflammation values, pointing at an inter-cellular mechanism. Secondly we showed that, in a co-culture of Hela cells and hepatocytes, cells are not randomly arranged, and that their relative positioning cannot be fully explained by a null model considering the size differences between the two cell types. Thus, PySpacell offers an approach to reduce the bias introduced by irregular shaped and randomly spaced objects as cells are, especially in culture. Furthermore, while Stahl and collaborators' data lay on a regular grid (Stahl et al. 2016), we demonstrated that the choice of neighborhood in PySpacell allows for a dedicated analysis of biological experiments. Finally, PySpacell provides a visualization of neighborhood and local spatial autocorrelation directly on the cell image which is useful for rapid in situ investigation.

We chose to model the cells with a marked point process, with each cell represented by a point. It makes biological sense to consider per cell feature values rather than pixels values directly because pixel size can be of various scales. Indeed, this point is valid for many assays: a cell is a biologically relevant unit, and for example averaging the lipid droplets fluorescence for the hepatocytes makes sense because the lipid droplets can be distributed in the whole cytoplasm. Edsgärd and collaborators also chose to model cell locations as a marked point process. Moreover they chose four summary statistics (E-mark, V-mark, Stoyan's mark correlation and mark variogram) that they finally aggregate into a single

feature (Edsgärd et al. 2018). In fact, marked point processes have been very extensively studied in Geography and Ecology (Szmyt 2014, ArcGIS software), and many techniques have been developed and refined for years. PySpacell provides computation of standard measurements as spatial autocorrelation and Ripley's functions without aggregating them in order to favor interpretability. In addition, Edsgärd's measures can only account for numerical marks, while PySpacell allows for detecting pattern from both quantitative and categorical features. The toolbox we propose is more flexible in this regard, because it can take as input any kind of feature as long as this feature can be computed per cell. Therefore, the traditional fluorescence features can conveniently be augmented with additional information such as MALDI or expression data from sequential FISH.

During the development of pySpacell, we thought to include multivariate methods to match the output of software program such as CellProfiler. In practice, those output features end up to be often highly redundant and necessitate an intermediary processing. A literature search (Dray and Jombart 2011) showed that most of spatial multivariate approaches rely on a modified principal component analysis (PCA) in order to compress the information, often at the cost of lower interpretability. As such preprocessing (e.g. projection on the first PCA) can easily be performed with other Python packages (e.g. scikit-learn), we focused PySpacell exclusively on the computation of spatial statistics from a single readout. Finally, PySpacell was conceived to ease spatial analysis of cells by the community and therefore was made easily available through the package management system pip.

Acknowledgements

Funding

This work has received support under the program "Investissements d'Avenir" launched by the French Government and implemented by ANR with the references ANR-10-LABX-54 MEMOLIFE and ANR-10-IDEX-0001-02 PSL* Université Paris. The collaboration between the Computational Bioimaging and Bioinformatics group in Paris and the Spatial metabolomics group in Heidelberg has been supported by the "Deutscher Akademisher Austauschdienst" (DAAD) by way of financing a short term research stay for France Rose at the European Molecular Biology Laboratory (EMBL).

Bibliography

ArcGIS software. ESRI 2011. ArcGIS Desktop: Release 10. Redlands, CA: Environmental Systems Research Institute.

Chen, X., Love, J. C., Navin, N. E., Pachter, L., Stubbington, M. J. T., Svensson, V., … Teichmann, S. . Single-cell analysis at the threshold. *Nature Publishing Group*, *34*(11), 1111–1118. https://doi.org/10.1038/nbt.3721 (2016).

Dixon, P. M. Ripley's K function. Encyclopedia of Environmetrics 3, 1796–1803 (2002).

Dray, S. & Jombart, T. Revisiting Guerry's data: Introducing spatial constraints in multivariate analysis. Annals of Applied Statistics 5, 2278–2299. issn: 19326157 (2011).

Edsgärd, D., Johnsson, P. & Sandberg, R. Identification of spatial expression trends in single-cell gene expression data. 15. doi:10.1038/nmeth.4634 (2018).

Keren, L. et al. A Structured Tumor-Immune Microenvironment in Triple Negative Breast Cancer Revealed by Multiplexed Ion Beam Imaging. Cell 174, 1373–1387.e19. issn: 10974172 (2018).

Legendre, P. & Legendre, L. Structure functions 2nd Englis. Chapter 13. 2, 712–738. doi:10.1093/oxfordjournals.bjc.a046803 (Elsevier Science BV, Amsterdam, 1998).

Lubeck, E., & Cai, L. Single-cell systems biology by super-resolution imaging and combinatorial labeling. Nature methods, 9(7), 743 (2012).

Newman, M. E. Mixing patterns in networks. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 67, 13. issn:1063651X (2003).

Rappez, L. et al. Spatial single-cell profiling of intracellular metabolomes in situ. bioRxiv, 1–52 (2019).

Rohban, M. H., Singh, S. & Carpenter, A. E. Capturing single-cell heterogeneity via data fusion improves image-based profiling. bioRxiv Bioinformatics. doi:10.1101/328542. http://biorxiv.org/cgi/content/short/328542v1 (2018).

Snijder, B. et al. Population context determines cell-to-cell variability in endocytosis and virus infection. Nature 461, 520–523. issn: 1476-4687 (2009).

Ståhl, P. L. et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science 353, 78–82. issn: 10959203 (2016).

Szmyt, J. Spatial statistics in ecological analysis: From indices to functions. Silva Fennica 48. issn: 22424075. doi:10.14214/sf.1008 (2014).

Svensson, V., Teichmann, S. A. & Stegle, O. SpatialDE: identification of spatially variable genes. 15. doi:10.1038/nmeth.4636 (2018).

Toth, T. et al. Environmental properties of cells improve machine learning-based phenotype recognition accuracy. Scientific Reports 8, 1–9. issn: 20452322 (2018).

Wang, X. et al. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. Science 361 (2018).

Zhu, Q., Shah, S., Dries, R., Cai, L. & Yuan, G.-c. Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data. Nature Publishing Group. issn: 1087-0156. doi:10.1038/nbt.4260.http://dx.doi.org/10.1038/nbt.4260 (2018).

5.3 Conclusion

5.3.1 The importance of the null model in statistical tests

The importance of the null model is crucial in any statistical test. It shapes the question that can be asked and thus answered. Indeed, when the null hypothesis of a test is rejected, the only conclusion that can be drawn is that the data cannot be explained by the process modeled by the null hypothesis, yet other processes could explain them. In the context of spatial statistics, there are mainly one null model. If the research questions are about the point spatial distribution, then the favored model will be Complete Spatial Randomness (CSR) [157]. This model takes one parameter, the intensity λ. The number of points drawn follows a Poisson law of parameter $A * \lambda$, with A the studied area [215]. Then points are taken independently according to a uniform 2D distribution. This null model is almost always used even when the data show signs of non randomness [167].

However, if points are labeled with categories, then two usual null hypotheses can be defined [157]. The first one hypothesizes the independence between the processes of each type, and thus allows to ask questions about the interaction of the processes. Under this hypothesis, Ripley's cross functions would all follow the CSR model with different intensities. The second hypothesis takes a different point of view: it considers random labeling, i.e. interchangeable point types. This approach asks questions about the process that assigns labels to points. Under this hypothesis, Ripley's cross functions are all equal. Departure from the random labeling hypothesis can be then examined using pairwise differences between K cross-functions. This last viewpoint is the one we favored for *PySpacell*, because we were mostly interested in the labeling process. Nonetheless, we could have complexified our null model by incorporating a step were data points are randomly repositioned, and it will be discussed in the next subsection

5.3.2 Null model simulations

For the case of the co-culture image, the cell spatial distribution does not follow the CSR model. Indeed when the K function is plotted for the cell centroids, the data curve is above the one obtained under CSR hypothesis (cf Figure 5.2 black and red curves). This difference means that the data points are more clustered than expected under the CSR hypothesis. Some processes, like the Neyman-Scott or the Cox processes, are able to model such clustering [152, 157]. For the Neyman-Scott process, the analytic formula for K function was derived and equals [215]: $K(t) = \pi * t^2 + (1 - exp(-t^2/4\sigma^2))/\rho$, where σ and ρ are the parameters of the Neyman-Scott model.

Neyman-Scott simulations

To model the clustering behavior of cells, I fitted the K curve to the analytic Neyman-Scott formula to obtain estimated values for σ and ρ. Then I drew new positioned data points with the following procedure: first, parent points are drawn from an homogeneous Poisson process with intensity ρ; then each parent point i generates a random number of offspring points N_i according to a Poisson law with parameter m, and the location of each offspring relative to the parent point follows a Gaussian distribution with 0 mean and σ standard

deviation in both x and y directions. I took m equal to N (the true number of cells) divided by ρ (average number of parent points). It is important to note that the parent points are not conserved in the output simulated points.

The K function fitting to Neyman-Scott model is good (cf Figure 5.2 a.), however the nearest neighbor distances of the simulated points do not match the nearest neighbor distance distribution of the true data points, especially for the small distances (cf Figure 5.2 b.). This observation is understandable, as cells are not points but occupy a non-null area, even if this trend did not show on the chosen scales displayed on the K function plot (cf Figure 5.2 a.). So at short distances, the data should be modeled with a soft-core or a hard-core process, and at medium distances, the data should be modeled under a clustering process. With the extent of my literature search, I did not find a good and simple model that combines both effects and that is well documented, for example with an analytic K function formula [146].

Another possibility would have been to get simulated points spatial distribution through an ad hoc process. For example, we could have iteratively generated points that match the distribution of nearest neighbors or overall distributions. We did not choose to follow this path, because generating such a distribution takes an exponentially growing amount of time as the cells are getting more confluent, and because searching the parameter space to evaluate the distance of inhibition or the probability to reject a point would have also increased greatly the complexity of the computed statistics via the toolbox.

5.3.3 A compromise between complex simulations and the simple random shuffling model

To sum up, in the case of cultured cells, it is hard to find a simple null model that is better than the random shuffling, as it is especially hard to mimic the positioning of cells. In the study case of hepatocytes, random shuffling is a suitable model, as all the cells are of same type and similar shape and size. However, for the study case of co-culture between Hela cells and fibroblasts, a blind random label shuffling is questionable as the two cell types do not share the same reference size or shape. Indeed with the random label shuffling, it is not possible to discriminate between the two following hypotheses: cells are clustered with cells of the same type, or cells are positioned this way because there are groups of small rounded cells and groups of elongated cells.

As described in the paper, we wanted to minimize the bias caused by blind random label shuffling: indeed in that process, a small rounded cell can be relabeled as a fibroblast even if its morphological characteristics would not physically allow a fibroblast at this position. This blind relabeling can have consequences on the number of neighbors the cell has at this position, and thus on the spatial statistics. So to avoid relabeling a cell as a fibroblast where the position does not allow it, I implemented in *PySpacell* a tool that checks the surroundings of a cell before assigning it a new label (cf Figure 3 of the paper). This option allows the user to set the number of nearest neighbor distances to take into account when assigning a new label.

The Figure 5.3 shows the impact of the choice of the number of controlled dimensions. The panel a. of Figure 5.3

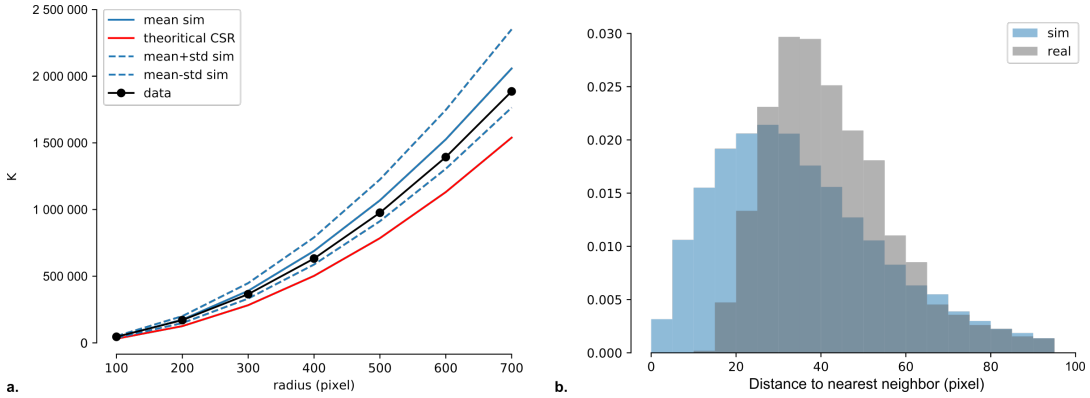


Figure 5.2: **Ripley's K and nearest neighbor distance distribution for Neyman-Scott simulations and the data. a.** Ripley's K computed for different neighborhood matrices with increasing euclidean distance: x-axis in pixels. "sim" is short for Neyman-Scott simulations. 999 permutations were computed, hence only the mean, the mean+std (standard deviation) and the mean-std are plotted. Theoritical CSR corresponds to Complete Spatial Randomness, $\pi * radius^2$. **b.** The distributions of distances between cell's nearest neighbors are displayed for the data (grey) and for a randomly chosen Neyman-Scott simulation (blue).

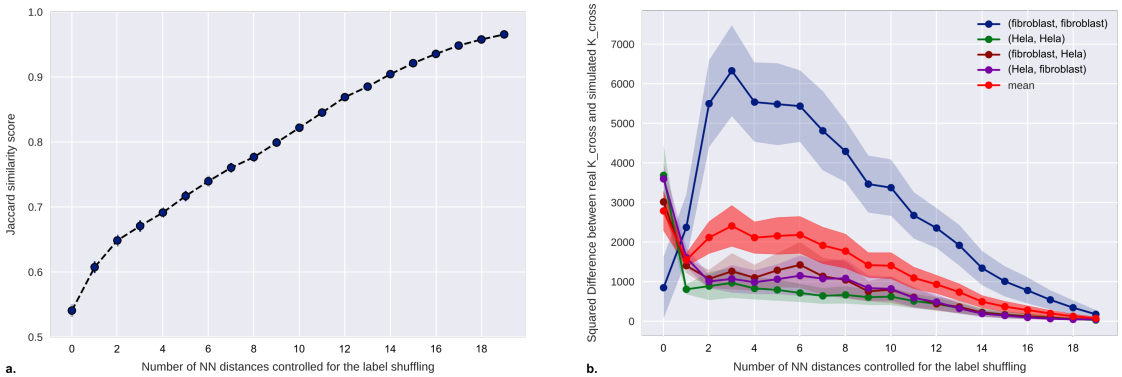


Figure 5.3: **Influence of the number of dimensions controlled for the label shuffling. a.** Jaccard similarity score between the original labels and the shuffled ones as a function of the number of nearest distances (nc_{NN}) controlled for the label shuffling. **b.** Squared difference between the real value of Ripley's cross-functions and the value obtained after label shuffling (denominated simulated K_cross). 100 permutations were computed for each value of nc_{NN}. The solid line corresponds to the mean, and the colored interval to the values between the mean-std (standard deviation) and the mean+std of the permutations.

represents the Jaccard index between the list of true labels and the list of shuffled labels, i.e. the percentage of cells that do not see their label changing in the shuffling process. It shows that when the controlled number of nearest neighbor distances nc_{NN} increases, the number of cells that keep the same label increases. This observation confirms that when matching the nearest neighbor distance (NN) distribution, the type of label that can be re-assigned for some cells with very close or very far neighbors is constrained. The panel b. of Figure 5.3 represents the sum of the squared differences between the true K cross-function and the K cross-function computed after the random shuffling, per cross-function and their average. This plot (cf Figure 5.3 b.) shows a general decrease of the difference between the true and the simulated cross-functions from the blind random shuffling ($nc_{NN} = 0$) to the first controlled nearest neighbor distribution ($nc_{NN} = 1$). However when $nc_{NN} > 1$, the difference between the true and the simulated cross-functions increases again. This secondary increase is mostly due to the cross-function $K_{(fibroblast,fibroblast)}$.

While the underlying process is not completely clear, looking at the shuffled labels' pattern directly on the image might give some intuition (cf Figure 5.4 a.). It seems that at $nc_{NN} = 0$, the labels are completely mixed (cf first panel of Figure 5.4 a.). At $nc_{NN} = 1$ and $nc_{NN} = 2$ (cf second and third panels of Figure 5.4 a.), the red dots are starting to regroup but some green dots are located on the outer part of the groups of reds and some in the middle of them, causing a decrease of the cross-function $L_{(fibroblast,fibroblast)}$ values (cf Figure 5.4 b.), and by way of consequence an increase of the difference between the real and simulated cross-functions (cf Figure 5.3 b.). At $nc_{NN} = 15$, most of the green dots have been pushed on the outer part of the more dense red clusters, it results in a higher proximity of the green dots, and an overall increase of the cross-function $L_{(fibroblast,fibroblast)}$ values.

Considering the average difference between simulated and real K cross-functions (cf Figure 5.3, red curve), we chose to show only the data for the first NN distance controlled in the body of the paper because it improved the model without applying two many constraints.

I chose not to go further with the null model point positioning simulations presented in the previous subsection, because of their computational complexity but also because generating points with the same interpoint distance distribution does not solve the problem of finding a way to assign cell types during the label shuffling.

5.3.4 The difference between cultured cells and tissue

PySpacell is technically usable on both cultured cell data and tissue data. Indeed, as long as tissue data are segmented and features collected per cell, all the *PySpacell* tests can be run. Once again, the question of the null model is crucial here. In the case of an homogeneous tissue, the feature values can be randomly shuffled. Although, only specific parts of tissues are homogeneous, and usually the benefit of studying a biological process on a tissue is to get closer to an actual living system. This implies some heterogeneity in the tissue, especially heterogeneity between cells. In this case, a blind shuffling will introduce a bias. For example, blind shuffling could exchange feature values between a

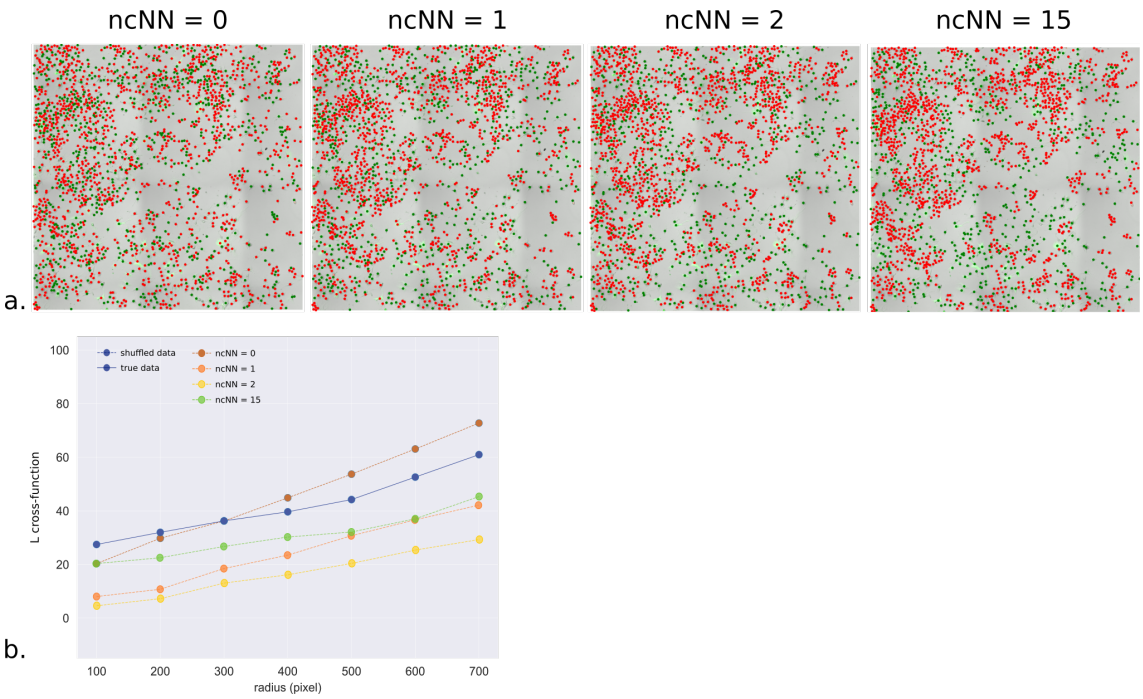


Figure 5.4: **Cells relabeling in function of the controlled number of nearest neighbor distances. a.** Overlay of the co-culture image with the re-assigned labels after a shuffling for 0, 1, 2 and 15 controlled nearest neighbor distances nc_{NN}. **b.** L cross-function $L_{(fibroblast,fibroblast)}$ (variance stabilized version of $K_{(fibroblast,fibroblast)}$) in the true data case, and in the shuffled cases with 0, 1, 2, and 15 controlled nearest neighbor distances nc_{NN}. In this figure, only one example of shuffling is shown for each value of nc_{NN}.

cell with a small volume and a cell with a bigger volume. Then concluding that there is a non-random spatial pattern for the feature could be biased by the underlying spatial non-homogeneity of cell sizes. In more details, in the case of a feature strictly correlated to the cell volume, and of a non-random spatial pattern of the cell volume, then a spatial statistical test for the feature would be significant.

Elise Laruelle, a post-doctoral researcher in our team, is currently working on these problematics: how to generate randomized tissue images, from an image of a segmented tissue, while preserving the shapes of the cells. She proceeds through an iterative process, moving cells step by step. In a few words, cells are modeled via a center and an ellipse. For each iteration, the center is first moved to be at the barycenter of the cell shape, then the cell borders are reconstructed to meet at best all the cells' shape requirements. The number of iterations necessary to create a new random image depends on the characteristics of the source image.

The spatial analysis of cell images is just at its beginning, and I hope that *PySpacell* will help biologists to ask and answer questions on the spatial randomness of their data. *PySpacell* is designed to be a first step towards this goal. To go further, we think that a proper null model should then be defined for each purpose.

Chapter 6

Conclusion

In this thesis, I focused on studying cellular heterogeneity grasped from microscopy images. For this study, I used several datasets: diverse high-content screens on patient-derived cell lines tested with compounds from the Prestwick library, the publically available BBBC021 screen on breast cancer cells treated with anti-cancer compounds [202], and fluorescence images combined with a spatial mass spectrometry modality. I considered the heterogeneity between cells, as cells can be considered as basic biological entities. Many biological phenomena are regulated at the level of a cell. Also several intra-cellular components are related to each other, and morphological information from one or another can be redundant [216]. This is what we illustrated when removing the actin and the tubulin fluorescence channels and computing the MOA accuracy (cf Chapter 3): phenotypic information from different cellular compartments can be redundant.

Image-based cellular heterogeneity can be seen at different levels: between cell lines, between treatments on a given cell line, and between different spatial neighborhoods on the same image. The morphological heterogeneity between cell lines is at the source of the increase of target prediction performance when combining data from diverse cell lines (cf Chapter 3). Inside a cell line treated with the same compound, phenotypic heterogeneity can be detected in the form of subpopulations. However, for the BBBC021 dataset, expressing treatment effect on cells with cell subpopulations does not guarantee improvement of MOA prediction. Nonetheless, a few clustering methods for subpopulations give good and similar results in terms of MOA prediction (cf Chapter 4). I approached the study of non-random spatial organization of subpopulations inside an image with cell planar graphs. I observed that they cannot be efficiently analyzed with common graph degree- and node-based measures. Yet direct comparison of image cell graphs via graph kernels can improve the MOA prediction under certain conditions (cf Chapter 4).

Downscaling the amount of images, I implemented and tested exploratory spatial statistics in a *Python* module, PySpacell (cf Chapter 5). PySpacell is the first dedicated spatial statistical analysis package for images. Cells are supplied with properties, like quantitative (level of a fluorescent marker) or qualitative (cell type) feature. The spatial organization of these properties can be tested for randomness with respect to the cell graph. The Python module is available for download on *github* [1]. With the help of PySpacell, I showed that there is a spatial clustering of similar values of lipids accumulation close to each other, and this spatial clustering is more important when considering links between cells that are touching rather than links based on Euclidean distance from cell centroids. In a second example, I showed that Hela cells and fibroblasts do not randomly mix when seeded together.

[1] https://github.com/biocompibens/pySpacell

Chapter 7

Discussion

7.1 The limits and potentialities of cell-based assays

7.1.1 BBBC021 is a model dataset for classical high-content screening

Cell-based assays are usually carried on adherent cells seeded in multi-well plates. Treatments are then applied in a parallel fashion. In order to proceed to drug profiling, metadata need to be collected on at least some compounds, to identify mechanisms of action or targets of unknown compounds, to perform lead hopping, or to enrich small molecule library [25]. For all these purposes, the general idea is to propagate known functional annotations to unknown compounds.

Because it possesses MOA annotations and because it is easily accessible (downloadable online at [201]), the BBBC021 dataset is a model dataset for drug profiling. This dataset is of consequent size with 3 general cell-labeling fluorescent channels, 103 treatments classified in 13 MOAs, and more than 2,000 images. Such a dataset is rare: indeed gathering functional annotations is time-consuming and most of screens use one fluorophore or very specific ones [20].

It has become a model dataset in that many methods have been tested on it [31, 34, 89, 202–204], hence it eases comparison of methods, as discussed in Chapter 3. However the BBBC021 has its limitations: it uses the MCF-7 cell line in a 2D context and its MOA classification may suffer some pitfalls.

7.1.2 MOA annotations are necessary but bear pitfalls

Although having MOA annotations is a key point of the BBBC021 dataset, the ones available are corresponding to different levels of annotation precision. Indeed, some are defined at the level of the protein that the drug targets (e.g. Eg5 inhibition), at the level of a cellular component (e.g. actin disruptors), or at the level of a more general cellular pathway [217]. These categories are usually defined at a level where differences can be seen on the screen images [217]. For the BBBC021 dataset, either the choice of compounds which give striking phenotypes, or the fine-tuned choice of the MOA category levels, allows an almost perfect accuracy score: the best methods reach over 90%. However for other datasets, like the one used in our paper [200] (cf Chapter 3), or other papers [217], the reached prediction accuracies are quite low - around 25%-30%. This may be due to unsuitable fluorescent markers, to chosen compounds, or to the prediction task itself, namely a larger number of categories than for the BBBC021 dataset [200] or a more difficult categories' separation [217]

Moreover, gathering functional annotations is not only time-consuming but can also be subjective. Indeed when searching online databases like chEMBL [218], PubCHEM [219], or PHAROS [220], different levels of information for different compounds are usually found: some are very well documented while some have very few informational content available. Also for some compounds several targets are listed, whereas for some others, none or one is reported. When creating a set of functional annotations for one dataset, these information needs to be evened out: for example, as we did for the Prestwick library, we chose to report

only a primary target. Unfortunately for some compounds, several targets are listed with no hint on which one might be the primary one, then two possibilities are to be chosen from: removing the compound from the dataset or randomly choosing one of the potential targets as the main one.

7.1.3 The choice of cell line(s) can improve data quality

In the preliminary optimization for a HCS assay, the choice of the cell line can be critical. Indeed by its characteristics, its morphology, its RNA and protein pools, the chosen cell line may display or not visible changes once treated [100]. When the potential range of tested compounds is large, one cell line might not be sensitive enough for all types of effects, all mechanisms of actions [100, 217]. Hence the choice of multiple cell lines can be a better option to obtain a larger set of phenotypic modifications. In Kang and coworkers [100], a method to systematically identify optimal reporter cell lines for sets of compounds was proposed. In the late 1980s, both in the United States and in Japan, panels of cell lines were created to represent the diversity found in each type of cancer, as a response to the poor results of bench-to-bedside translation [221]. These combinations of cell lines per type of cancer could be used routinely in HCS to widen the explored genetic and transcriptomic landscape, combined with a multi-cell-line data analysis method like ours (cf Chapter 3) or others [91, 217].

Overall the intrinsic quality of classical cell lines can be questioned. Some processes are inherently altered in usual laboratory cell lines and altered further by successive passages. Cross-contamination is largely suspected as well [221, 222]. One answer to the limitations of classical cell lines is the use of more recent cell lines that were banked at lower passage and with less stringent culture conditions [221], or patient-derived cell strains, making them more clinically relevant. Another pitfall that can arise with any cell handling is their diversification and divergence: indeed it has been shown that cell lines identically labeled in different laboratories differ subsequently [223]. Namely, Ben-David and collaborators found that "between the two sources of cell line data, a median of 19% of the detected non-silent mutations and 26% of gene copy number alterations (CNAs) were present in one data set but not the other." [222] This diversity has consequences on cellular drug response: "In a screen of 321 compounds, the drug response of the different MCF7 strains was highly variable: among the 55 compounds that inhibited the growth of at least one strain by more than 50%, 48 of these showed $<$20% growth inhibition in at least one other strain." [221] The generalization of fast sequencing could limit the variability between assays but also bring new information to integrate data: with tested cell lines quite close to each other, phenotypic diversity would be observed but probably with a high overlap, and this complex pattern of differences could be related to genetic mutations.

7.1.4 3D culture assays are developed for drug screening

Another lead to improve *in vitro* disease models is to select and grow cells in 3D. Indeed it is known that adherent cancer cells grown in 2D have a deregulated cell cycle, with a much higher proliferation rate than what has been observed in tumors *in vivo* [21]. By culturing them in monolayer on plastic,

strains are selected for these conditions and they do not display real tumor-like characteristics, genetic ones as well as phenotypic ones [21, 221]. In particular the characteristics of stiffness and crowdedness are far from *in vivo* situations. These limitations may impair the process of drug screening: for example, anti-cancer drugs selected from 2D experiments based on their ability to stop proliferation do little in real conditions [21].

Efforts to develop cellular models in 3D have been booming recently. Modalities of different levels of complexity are being improved: from the more simple co-cultures, spheroids, micro-tissues, to the more complex organoids [21]. Distinctions are made mostly on the number of cell types which cohabit and their co-organization, and how well the tissue organization is reproduced. Main techniques involve culture in anchorage, hanging drop method and ultra-low attachment plates. Microfluidics devices are also skillfully designed to handle multiple cell lines or to create ad hoc culture chambers [224, 225]. They are believed to better reproduce *in vivo* systems.

However some major difficulties stand on their way to larger-scale higher-throughput assays: the liquid handling for cells and extra-cellular matrix components is hard to automatize, the sample preparation including the fixation and the fluorescent labeling might be impaired by the 3D structure, the imaging modalities have to be improved for a complete imaging of the spheroid or the organoid, ... [21] All these parameters affect the cost, the speed and the scale at which experiments can be carried. Also as these 3D methodologies are fairly recent, many of them coexist and no common standard protocols have emerged yet. Hence the variation between assays and between laboratories is currently higher than the one observed for 2D cultures [21]. In summary, 3D cultures display a great potential for drug screening, but remain a young technology and present many technical challenges before they could be routinely implemented for HCS.

The type of data analysis required by these new methodologies, would probably be twofold: first, in continuity with current data analyses based on segmentation, more numerous or more complex per-cell features can be exploited, and secondly, features based on the newly available dimension, e.g. extra-cellular matrix permeability, quantification of the multi-cellular organization, etc. Yet the lower axial resolution compared to the planar one is a known pitfall of 3D data, and can impair the computed per-cell features. Deep learning approaches will certainly get a piece of the pie, as they can overcome problems of segmentation, of lower axial resolution as long as the training data are numerous and consistent.

I believe 3D collected data would display in general an increased diversity and morphological analysis assisted by neighborhood and high-level organization analysis.

7.2 Finding subpopulations: what for?

7.2.1 Our benchmark of clustering methods for HCS may be the first

I tested four methods, k-means, k-means per channel, flowSOM and PhenoGraph, on several feature matrices extracted from the BBBC021 dataset. The obtained results are compatible and comparable to the ones obtained by Ljosa

and collaborators [34] as the set of original features is the same. It is the first time, to our knowledge, that some unsupervised subpopulation methods are tested side by side on HCS data. Indeed, Ljosa and coworkers compared unsupervised drug profiling methods on the BBBC021 dataset, however only one of them targets subpopulations, the Gaussian Mixture Model [34]. Reisen and collaborators presented a consequent benchmark of similarity measures and feature selection methods in 2013 [134]. These methods were tested for the separation between negative and positive controls and for the separation between multiple phenotypes. Again, no subpopulation algorithms were used. In a preprint published by Weber and Robinson in 2016 [50], numerous clustering methods developed for single-cell flow and mass cytometry were benchmarked. It is a more common analysis to group cells into subpopulations in cytometry than it is in HCS. Indeed since 2010, a consortium "Flow Cytometry: Critical Assessment of Population Identification Methods" has organized regular challenges to evaluate rigorously the performance of various methods [50, 226]. This is not currently the case for HCS.

7.2.2 Subpopulations are routinely looked for in flow cytometry data, and more recently single-cell genetic expression data

Flow cytometry is widely used to study, identify, and quantify cell subpopulations. It gathers one fluorescence value per tagged protein per cell. The traditional analysis for flow cytometry data involves manual gating: an operator looks at 2-dimensional scatter plots of the fluorescence values per cell, groups cells by similarity, and identifies foregone subpopulations [50]. The tagged proteins are usually well thought to identify known subpopulations, like disease biomarkers, key protein abundances or position. However, lately the number of usable parallel fluorophores has increased, and manual gating is less and less an option, as the number of scatterplots to visualize and choose from increases exponentially [50, 52]. Hence clustering methods have been developed to supplement the operator in his/her work. After subpopulations are determined, a possibility is to sort the cells and study each subpopulation in isolation. Physical subpopulation extraction is more complex for microscopy: indeed cells need to be alive during the observation - this usually requires a special incubation chamber adapted to the microscope -, then the positions of a few wanted cells are detected to proceed to microdissection (CAMI - computer assisted microdissection [227]). In opposite to the flow cytometry, these microdissections are far from a routine method.

The advances in single-cell genetic expression methods, notably single-cell RNA sequencing, brought lately single-cell data, in which heterogeneity and subpopulations can also be studied [47]. The use of lineage softwares as long as clustering methods are frequent [145, 228]. The main idea of lineage algorithms is to fit the data into a low dimensional space, then link data points by proximity, and convey a meaning with these links by giving them a directionality called pseudo-time. The first lineage algorithm published, Monocl [229], uses independent-component analysis for the dimension reduction step, and a minimum spanning tree to connect the cells. It was first applied in the context of cell differentiation: there the pseudo-time represents the trajectory of cells from a non-differentiated state to differentiated one(s).

This type of algorithms could be used for HCS single-cell data. The idea of trajectories has already been stated in the past by several groups [29, 30, 108, 230].

Loo and coworkers used SVMs to represent the separation between treated and untreated cells with one vector [29]. The evolution of this multi-dimensional vector was studied as the drug dose increased. Some concentrations were grouped together, discriminating ranges of action for the compounds. This grouping is done at the scale of the population, not subpopulations. It is a way to visualize the effect of the treatments but does not link diverging or converging cell fates.

Yin and collaborators detected relatively invariant phenotypes, which are called attractor states, linked by transition states [108]. Their approach has not been tested on other datasets, and we do not know if their modelization of the morphological landscapes is applicable to a large spectrum of situations. However, the underlying hypothesis of a continuous deformation of the phenotypic space from an "original" state, mostly populating the negative control, to diversified phenotypic states reached under the influence of treatments is hard to validate at first. This will probably remain a complex issue as subpopulations cannot be isolated for further biological testing of these hypotheses, and as the space of subpopulations cannot be easily deduced from the MOA or treatment space. Indeed several decompositions in subpopulations can be validated in terms of MOA classification without bearing the correct information about the cellular biological processes.

The study of Gut and collaborators [230] is the one based on images resembling the most trajectory analyses found in genomics. Indeed, they built cell-cycle trajectories from images of fixed cells. The idea was to understand the impact of the cell-cycle phase on other biological processes and cell-to-cell variability. Along with the study by Yin and collaborators [108], this method reconstructs single-cell trajectories. Yet contrary to the first two examples [29, 108], the trajectory is based on a known biological progression of cells and is a preliminary step for further cell-to-cell variability analyses. This approach overcomes the main limit of Yin et al. [108], but is not blindly applicable to other biological processes.

To make a larger use of single-cell trajectory methods from image data, studies need to be based on already characterize processes like cell-cycle or differentiation, using designed fluorescent markers. For now, no proof of concept study has shown it to be possible in the general context of drug perturbation, and this conceptual step would require substantial downstream biological testing.

7.2.3 Making sense of subpopulations may require to find relationships between them

In the methods I tested to find subpopulations, most of them gave good MOA prediction accuracy results, but did not surpass the accuracy found by the simplest method that Ljosa et al. tested, the mean vector [34]. As we discussed previously, it might be because the MOA are easy to distinguish in this case, and more complex methods may not bring additional information. Though clustering methods may be of great help on other datasets.

A way to make use of subpopulations may be to enrich the available metadata and link them with genetic data like

gene expression networks. Barabasi and coworkers linked gene or protein-protein interaction graphs with diseases, and studied among other things how drugs can be repurposed by finding shortest paths between a drug target and a gene known to be implicated in a disease. These gene-disease networks are created from genetic associations and comorbidity data. In a gene network, the protein interactions are assumed to mechanistically code for the disease from a network perspective. The better suited the genetic data is, the better the prediction would be: for example, gene expression patterns differ from tissue to tissue, or from cell type to cell type and knowing the gene network corresponding to the tissue might make the difference between a working repurposing and one that does not work. Another use of these networks is to suggest new protein that might be associated with the disease, hence new potential targets.

The subpopulations might complement this network. In the previously described gene graph, a disease is associated with a number of gene nodes, and a drug with a priori one node (its primary target). Now if we add the subpopulations, they can be associated with several drugs, hence several gene nodes, it could be checked if they are consistently associated with the same nodes and if they form topological clusters in the graph, i.e. connected components. For each subpopulation, assumptions could be made based on the associated cluster of genes on which pathway is impaired. Then the description of a treatment could be enriched and complexified by building it from the different gene clusters associated with the major subpopulations the treatment is causing to cells. Few studies have started to explore this relationship [231, 232].

In the context of an RNAi screen, Evans and collaborators used phenotypic information to construct gene interactions [231]. In practice, subpopulations were detected with a semi-supervised method. The gene perturbations are then clustered into "phenoclusters" based on their associated subpopulations. They assumed that genes being grouped in these phenoclusters have a high chance to act in the same pathway. However this method has caveats, as there is no way to infer any directionality in the gene interactions, also a blind decision needs to be made on the level at which the clustering needs to happen. The authors proposed to combine that with prior knowledge, as co-expression or protein-protein interaction data.

Reisen and coworkers used another approach for a drug screen. Subpopulations were grouped via a similarity network before making supra-clusters [232]. The supra-clusters were then "analyzed for enrichment of individual targets and gene sets". Enrichments were observed for 11 of the 84 supra-clusters. The possible follow-up could be to suggest gene sets or pathway on which unknown drugs act, enhancing the quality of the drug profiling. I suggest to carry this type of analysis at a larger scale, for example with the compounds from the Prestwick library as we already have target-compound associations (cf Chapter 3).

7.3 Spatial analysis: a way for a more complete data analysis?

7.3.1 Spatial analysis in cell images stays an unexploited source of information

Spatial relationships between cells or polarity inside cells is usually overlooked. The content of a microscopy image is high and most methods are ignoring most of it and applying a harsh dimension reduction one way or another [20]. Nonetheless results from the literature show the importance of spatial information even in cultured cells [60, 86, 150, 204]. Some biological processes cause large spatial supra-cellular organization. The example of viral infection showed by Snijder and coworkers [13] might be one of them. It is reasonable to hypothesize that division, lineage and cohabitation act on major cell processes, hence making two neighboring or two affiliated cells more alike in general than two cells taken randomly.

The spatial statistics implemented in pySpacell can help discriminating which biological process is more spatially organized, as they allow to perform exploratory analysis. Once again, different cell types and strains could react differently for different fundamental processes (cf Chapter 5).

7.3.2 Spatial aware methods have started to be tested

Few neighboring features are generally used: for example in CellProfiler, the distances to the n closest neighbors and their identity are available in one click. However more refined features, like the ones used in Snijder et al. [13, 86], e.g. local cell density, population size, and cell islet edge membership, or the ones used in Toth et al. [204] and Rohban et al. [233], i.e. the concatenated averaged feature values and other moments from the neighboring cells require for now manual implementation. The later approaches seem largely applicable and do not depend on the cell line or biological process involved.

Furthermore, contrary to the per-cell features, methods working on unsegmented images can leverage the supra-cellular spatial information. Previous methods, like WND-CHARM [77] or PhenoRipper [76], extract features defined closely to the ones used for segmented cells, based on intensity and texture. They are applied in contexts where the segmentation is difficult to achieve, for example cells with many filopodia or neurons, or when the needed information is quite basic but time is key. Yet these methods are not widely used.

Newer methods using convolutional neural networks can be fed whole images or parts of images as well and perform unquestionably well [31, 89, 203]. As they work on whole images and build customized features adapted to the task, they might select features that represent supra-cellular patterns. It is known that in the first layers of convolutional neural networks applied on images, simple features are combined in successive layers to form more complex and organized ones. However, to my knowledge, there has not been a thorough analysis of these intermediate features to see if there is a significant benefit of this supra-cellular level. With importance, saliency, activation map or other similar method, it is possible to visualize on the original image what groups of pixels have mainly contributed to the classifier decision. If the important pixel areas are located on cell borders or systematically on groups of cells, it might mean that indeed key features are not per-cell ones but necessitate a larger viewpoint.

My proposed approach based on graph

kernels (cf Chapter 5) is able to target this supra-cellular level and has proven to increase the prediction accuracy in some conditions. Although I did not explore all the extent of graph kernels possibilities, e.g. type of kernel, parameter space, other dataset, etc, I did not see a clear directionality for improvement or for next tests. I discussed in Chapter 4 the possible impact of the sparsity of the vector ϕ which stores the quantities of each type of node labels. When increasing the number of iterations of the algorithm, added labels are representing a growing neighborhood, and the length of the ϕ vector is increasing along. As the number of iterations is getting high, the number of neighbors pooled together is increasing as well, so the added label is more and more specific, and the chances to find them matching between images are dropping. To overcome this pitfall, I tried to decrease the number of subpopulations, but it compromises the quality of the MOA classification. However, from another point of view, one can also try to increase the size of the image or the number of cells per image. Then there would be no need to decrease abruptly the number of subpopulations and weaken the original MOA predictions. The computing complexity of the graph kernel should not be forgotten: actually Weisfehler-Lehman kernel is adapted to large graphs, but the number of images in a high-content screen is still high enough so it might be critical for some applications.

Another track for improvement is to weight the positions of the ϕ when doing the scalar product, then the later positions containing the more complex labels would have more emphasis. Then the kernel value will depend on the more global neighboring structure. It would be a good way to compare the importance of the different scales. Indeed without weights, there are higher numbers in the first positions of the ϕ vector so most of the decision is born by the individual cell subpopulation memberships.

In the result chapter 5, I also discussed the differences in prediction accuracy caused by the precise protocol of data pooling from images of the same well or of different replicates. For the graph kernel approach, I have not tried to average directly the ϕ vectors before computing the scalar product, but only to average the kernel values after the scalar product computation. It would be interesting to see the impact of this slight change. Also if the calculations of the ϕ vector and the scalar product are decoupled, similarity measure other than the scalar product could be tested (cf benchmark of similarity measure of [232]).

7.3.3 Spatial analysis applied to more relevant disease models carries promises

One approach to spatial analysis, graph kernels, could fit particularly well in the context of cultures with several cell types, as it is based on discrete categories of cells and as the interaction between diverse cell types might be more meaningful. Indeed, for co-cultures and 3D models, a greater diversity of phenotypes could be observed. The assay is making room for more diverse microenvironments and the full potential of cellular phenotypes. To study these assays, the need for morphological and in-depth analysis is required. Especially, the boundary between micro and macro heterogeneity as expressed by Gough and collaborators [35], needs probably to be soften. Through cell-cell interactions, cells can express a defined genetic identity - cell type - as well as variations of it.

A recent study [60] found that in brain cortex cells, some gene signatures reflect more the cell location rather than the cell type, assigned from cell type markers. This result shows that these distinctions might be reconsidered in relation to the context of the experiment.

In this context, the simple averaging of features per cell or even of features of neighboring cells will unlikely be sufficient to gather informative outcome of these experiments. Moreover the recent methodological work on subpopulations and cellular heterogeneity may be adequate for these new types of data [60, 204, 233, 234]. As these assays are getting biologically more relevant and allow for a more complex supra-cellular organization, the treatments may affect patterns of association of cell subtypes and other supra-cellular features.

The global trend for HCS assays evolves towards more precise and complete data. We will be able to see more differences and to quantify more sources of heterogeneity. The data is likely to become more frequently multiplexed with an increasing number of genes, metabolites and organelles followed at the same time, with a better spatial and time precision. It will probably all at once increase the content of HCS, and innovative data analysis methods will be required. Deep learning is giving a short answer to some of the HCS-derived tasks, like efficient treatment classification. However the need of directed method to study some biological processes is still significant to overcome the black box effect of convolutional neural networks.

List of abbreviations

AUC	Area Under the Curve
BMU	Best Matching Unit
CC	Connected Components (of a graph)
CNN	Convolutional Neural Network
CSR	Complete Spatial Randomness
CV	Cross-Validation
DAPI	4',6-DiAmidino-2-PhenylIndole
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNA	DeoxyriboNucleotide Acid
EM	Expectation-Maximization
FACS	Fluorescence-Activated Cell Sorting
GFP	Green Fluorescent Protein
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HCS	High Content Screening
HMRF	Hidden Markov Random Field
HTS	High Throughput Screening
i.i.d.	independent and identically distributed
kNN	k-Nearest Neighbors
KS	Kolmogorov–Smirnov
kSVM	kernel-Support Vector Machine
LDA	Linear Discriminant Analysis
LLE	Local Linear Embedding
LOO	Leave One Out
MALDI	Matrix-Assisted Laser Desorption Ionization
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MDS	Multi-Dimensional Scaling
MOA	Mechanism Of Action
MS	Mass Spectrometer
MST	Minimal Spanning Tree
MVA	Master Vision Apprentissage
NN	Nearest Neighbor
NP-hard	Non-deterministic Polynomial time-hard
PCA	Principal Component Analysis
RFS	Random Forest Scaling
ROC	Receiver Operating Characteristic
SOM	Self-Organizing Map
SPADE	Spanning-tree Progression Analysis of Density-normalized Events
sPCA	spatial Principal Component Analysis
SVD	Singular Vector Decomposition
SVM	Support Vector Machine
tSNE	t-distributed Stochastic Neighbor Embedding
WL	Weisfeiler-Lehman (graph kernel)

List of Figures

Bibliography

1. Marx, V. The big challenges of big data (2013).
2. *EBI. Our impact* www.ebi.ac.uk/about/our-impact (2019).
3. *Broad Bioimage Benchmark Collection* https://data.broadinstitute.org/bbbc/index.html (2019).
4. *Image Data Resource* idr.openmicroscopy.org/about/ (2019).
5. *Allen Brain Atlases and Data* www.brain-map.org (2019).
6. Eliceiri, K. W. *et al.* Biological imaging software tools. *Nat. Methods* **9,** 697–710. ISSN: 1548-7105 (2012).
7. *Google Dataset Search* toolbox.google.com/datasetsearch (2019).
8. Sacher, R., Stergiou, L. & Pelkmans, L. Lessons from genetics: interpreting complex phenotypes in RNAi screens. *Current Opinion in Cell Biology* **20,** 483–489. ISSN: 09550674 (2008).
9. Houle, D., Govindaraju, D. R. & Omholt, S. Phenomics: the next challenge. *Nature Reviews Genetics* **11,** 855–866. ISSN: 1471-0056 (2010).
10. Levsky, J. M. & Singer, R. H. Gene expression and the myth of the average cell. *Trends in Cell Biology* **13,** 4–6. ISSN: 09628924 (2003).
11. Ramsköld, D., Wang, E. T., Burge, C. B. & Sandberg, R. An abundance of ubiquitously expressed genes revealed by tissue transcriptome sequence data. *PLoS Comput. Biol.* **5,** e1000598 (2009).
12. Acar, M., Mettetal, J. T. & Van Oudenaarden, A. Stochastic switching as a survival strategy in fluctuating environments. *Nature Genetics* **40,** 471–475. ISSN: 10614036 (2008).
13. Snijder, B. *et al.* Population context determines cell-to-cell variability in endocytosis and virus infection. *Nature* **461,** 520–523. ISSN: 1476-4687 (2009).
14. Snijder, B. & Pelkmans, L. Origins of regulated cell-to-cell variability. *Nature reviews. Molecular cell biology* **12,** 119–125. ISSN: 1471-0072 (2011).
15. Bickle, M. The beautiful cell: High-content screening in drug discovery. *Analytical and Bioanalytical Chemistry* **398,** 219–226. ISSN: 16182642 (2010).
16. Schenone, M., Dančık, V., Wagner, B. K. & Clemons, P. A. Target identification and mechanism of action in chemical biology and drug discovery. *Nat. Chem. Biol.* **9,** 232–240 (2013).
17. Boutros, M., Heigwer, F. & Laufer, C. Microscopy-based high-content screening. *Cell* **163,** 1314–1325. ISSN: 10974172 (2015).

18. Swinney, D. C. & Anthony, J. How were new medicines discovered? *Nature Reviews Drug Discovery* **10,** 507–519. ISSN: 1474-1776 (July 2011).

19. Gregori-Puigjané, E. *et al.* Identifying mechanism-of-action targets for drugs and probes.

20. Singh, S., Carpenter, A. E. & Genovesio, A. Increasing the Content of High-Content Screening: An Overview. *Journal of biomolecular screening* **19.** ISSN: 1552-454X. doi:10.1177/1087057114528537 (2014).

21. Booij, T. H., Price, L. S. & Danen, E. H. J. 3D Cell-Based Assays for Drug Screens: Challenges in Imaging, Image Analysis, and High-Content Analysis. *SLAS DISCOVERY: Advancing Life Sciences R&D,* 247255521983008. ISSN: 2472-5552 (2019).

22. Feng, Y., Mitchison, T. J., Bender, A., Young, D. W. & Tallarico, J. A. Multi-parameter phenotypic profiling: using cellular effects to characterize small-molecule compounds. *Nat Rev Drug Discov* **8,** 567–578. ISSN: 1474-1776 (2009).

23. Danuser, G. Computer vision in cell biology. *Cell* **147,** 973–978 (2011).

24. Slack, M. D., Martinez, E. D., Wu, L. F. & Altschuler, S. J. Characterizing heterogeneous cellular responses to perturbations. *Proceedings of the National Academy of Sciences of the United States of America* **105,** 19306–11. ISSN: 1091-6490 (2008).

25. Caicedo, J. C., Singh, S. & Carpenter, A. E. Applications in image-based profiling of perturbations. *Current Opinion in Biotechnology* **39,** 134–142. ISSN: 18790429 (2016).

26. Carpenter, A. E. *et al.* CellProfiler: image analysis software for identifying and quantifying cell phenotypes. *Genome biology* **7,** R100. ISSN: 1465-6914 (Jan. 2006).

27. Schindelin, J. *et al.* Fiji: an open-source platform for biological-image analysis. *Nature methods* **9,** 676 (2012).

28. Fuchs, F. *et al.* Clustering phenotype populations by genome-wide RNAi and multiparametric imaging. *Molecular systems biology* **6,** 370. ISSN: 1744-4292 (2010).

29. Loo, L.-H., Wu, L. F. & Altschuler, S. J. Image-based multivariate profiling of drug responses from single cells. *Nature methods* **4,** 445–453. ISSN: 1548-7091 (2007).

30. Yin, Z. *et al.* A screen for morphological complexity identifies regulators of switch-like transitions between discrete cell shapes. *Nature cell biology* **15,** 860–71. ISSN: 1476-4679 (2013).

31. Godinez, W. J., Hossain, I., Lazic, S. E., Davies, J. W. & Zhang, X. A Multi-Scale Convolutional Neural Network for Phenotyping High-Content Cellular Images. *Bioinformatics.* ISSN: 1367-4803. doi:10.1093/bioinformatics/btx069. https://academic.oup.com/bioinformatics/article/2997285/A (2017).

32. Ando, D. M., McLean, C. & Berndl, M. Improving phenotypic measurements in high-content imaging screens. *bioRxiv,* 161422 (2017).

33. Caicedo, J. C. *et al.* Data-analysis strategies for image-based cell profiling. *Nature Methods* **14,** 849–863. ISSN: 15487105 (2017).

34. Ljosa, V. *et al.* Comparison of methods for image-based profiling of cellular morphological responses to small-molecule treatment. *J Biomol Screen* **18,** 1–16 (2013).

35. Gough, A. H. *et al.* Identifying and quantifying heterogeneity in high content analysis: Application of heterogeneity indices to drug discovery. *PLoS ONE* **9.** ISSN: 19326203. doi:10.1371/journal.pone.0102678 (2014).

36. Li, B. & You, L. Predictive power of cell-to-cell variability. *Quantitative Biology* **1,** 131–139 (2013).

37. Wernet, M. F. *et al.* Stochastic spineless expression creates the retinal mosaic for colour vision. *Nature* **440,** 174–180. ISSN: 14764687 (2006).

38. Altschuler, S. J. & Wu, L. F. Cellular Heterogeneity: Do Differences Make a Difference? *Cell* **141,** 559–563. ISSN: 00928674 (2010).

39. Ferrell, J. E. & Machleder, E. M. The biochemical basis of an all-or-none cell fate switch in Xenopus oocytes. *Science* **280,** 895–898 (1998).

40. Loo, L.-H. *et al.* An approach for extensibly profiling the molecular states of cellular subpopulations. *Nature Methods* **6,** 759–765. ISSN: 1548-7091 (2009).

41. Cohen, A. A. *et al.* Dynamic proteomics of individual cancer cells in response to a drug. *science* **322,** 1511–1516 (2008).

42. Gascoigne, K. E. & Taylor, S. S. Cancer cells display profound intra-and interline variation following prolonged exposure to antimitotic drugs. *Cancer cell* **14,** 111–122 (2008).

43. Sharma, S., Kelly, T. K. & Jones, P. A. Epigenetics in cancer. *Carcinogenesis* **31,** 27–36 (2010).

44. Brock, A., Chang, H. & Huang, S. Non-genetic heterogeneity—a mutation-independent driving force for the somatic evolution of tumours. *Nature Reviews Genetics* **10,** 336 (2009).

45. Spencer, S. L. Non-genetic origins of cell-to-cell variability in TRAIL-induced apoptosis. **459.** doi:10.1038/nature08012 (2009).

46. Campbell, L. L. & Polyak, K. Breast tumor heterogeneity: Cancer stem cells or clonal evolution? *Cell Cycle* **6,** 2332–2338. ISSN: 15514005 (2007).

47. Stuart, T. & Satija, R. Integrative single-cell analysis. *Nature Reviews Genetics.* ISSN: 14710064. doi:10.1038/s41576-019-0093-7. http://dx.doi.org/10.1038/s41576-019-0093-7 (2019).

48. Consortium, T. H. The Human Cell Atlas. White Paper. 1–15 (2014).

49. *10X Genomics* www.10xgenomics.com (2019).

50. Weber, L. M. & Robinson, M. D. Comparison of Clustering Methods for High-Dimensional Single-Cell Flow and Mass Cytometry Data. *bioRxiv,* 047613. ISSN: 15524922 (2016).

51. Stubbington, M. J. T., Rozenblatt-rosen, O. & Regev, A. Single-cell transcriptomics to explore the immune system in health and disease. **63,** 58–63 (2017).

52. Van Gassen, S. *et al.* FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry Part A* **87,** 636–645. ISSN: 15524930 (2015).

53. Bendall, S. C. *et al.* NIH Public Access. *Science* **332,** 687–696 (2012).

54. Wang, G., Moffitt, J. R. & Zhuang, X. Multiplexed imaging of high-density libraries of RNAs with MERFISH and expansion microscopy. *Scientific reports* **8,** 4847 (2018).

55. Codeluppi, S. *et al.* Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature methods* **15,** 932 (2018).

56. Wang, X. *et al.* Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science* **361,** eaat5691 (2018).

57. Shah, S., Lubeck, E., Zhou, W. & Cai, L. In Situ Transcription Profiling of Single Cells Reveals Spatial Organization of Cells in the Mouse Hippocampus. *Neuron* **92,** 342–357. ISSN: 10974199 (2016).

58. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics - supp data. *Science* **353,** 78–82. ISSN: 10959203 (2016).

59. Jansson, E. T., Comi, T. J., Rubakhin, S. S. & Sweedler, J. V. Single Cell Peptide Heterogeneity of Rat Islets of Langerhans. doi:10.1021/acschembio.6b00602 (2016).

60. Keren, L. *et al.* A Structured Tumor-Immune Microenvironment in Triple Negative Breast Cancer Revealed by Multiplexed Ion Beam Imaging. *Cell* **174,** 1373–1387.e19. ISSN: 10974172 (2018).

61. Rappez, L. *et al.* Spatial single-cell profiling of intracellular metabolomes in situ. *bioRxiv,* 1–52 (2019).

62. Singh, D. K. *et al.* Patterns of basal signaling heterogeneity can distinguish cellular populations with different drug sensitivities. *Molecular Systems Biology* **6,** 369. ISSN: 1744-4292 (2010).

63. Steininger, R. J. *et al.* On comparing heterogeneity across biomarkers. *Cytometry Part A* **87,** 558–567. ISSN: 15524930 (2015).

64. *Alan Brain Map. Alan Brain Atlases and Data* portal.brain-map.org/ (2019).

65. Bray, M.-A. *et al.* Cell Painting, a high-content image-based assay for morphological profiling using multiplexed fluorescent dyes. *Nat. Protoc.* **11,** 1757–1774 (2016).

66. Kankaanpää, P. *et al.* BioImageXD: an open, general-purpose and high-throughput image-processing platform. *Nature methods* **9,** 683 (2012).

67. De Chaumont, F. *et al.* Icy: an open bioimage informatics platform for extended reproducible research. *Nature methods* **9,** 690 (2012).

68. Rasband, W. *ImageJ* imagej.nih.gov/ij/ (2019).

69. Peng, H., Ruan, Z., Long, F., Simpson, J. H. & Myers, E. W. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology* **28,** 348 (2010).

70. Held, M. *et al.* CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nature methods* **7,** 747 (2010).

71. Sommer, C. & Gerlich, D. W. Machine learning in cell biology – teaching computers to recognize phenotypes. *Journal of Cell Sciience* **126,** 5529–5539. ISSN: 1477-9137 (2013).

72. Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools.* opencv.org (2000).

73. *ITK: The Insight Toolkit* itk.org (2019).

74. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of machine learning research* **12,** 2825–2830 (2011).

75. Kozak, K., Agrawal, A., Machuy, N. & Csucs, G. Data Mining Techniques in High Content Screening: A Survey. *Journal of Computer Science & Systems Biology* **2,** 219–239. ISSN: 09747230 (2009).

76. Rajaram, S., Pavie, B., Wu, L. F. & Altschuler, S. J. PhenoRipper: Software for rapidly profiling microscopy images. *Nature Methods* **9,** 635–637. ISSN: 15487091 (2012).

77. Orlov, N. *et al.* WND-CHARM: Multi-purpose image classification using compound image transforms. *Pattern recognition letters* **29,** 1684–1693 (2008).

78. Ando, D. M., McLean, C. & Berndl, M. Improving Phenotypic Measurements in High-Content Imaging Screens. *bioRxiv,* 161422 (2017).

79. Sommer, C., Hoefler, R., Samwer, M. & Gerlich, D. W. A deep learning and novelty detection framework for rapid phenotyping in high-content screening. *Molecular Biology of the Cell,* mbc.E17–05–0333. ISSN: 1059-1524 (2017).

80. Perlman, Z. E. *et al.* Multidimensional drug profiling by automated microscopy. *Science* **306,** 1194–1198 (2004).

81. Rose, L. T. *The End of Average* www.thestar.com/news/insight/2016/01/16/when-us-air-force-discovered-the-flaw-of-averages.html (HarperCollins Publishers Ltd, 2016).

82. Duda, R. O., Hart, P. E. & Stork, D. G. *Pattern Classification* 680. ISBN: 0471056693. doi:10.1007/BF01237942 (2001).

83. Neumann, B. *et al.* High-throughput RNAi screening by time-lapse imaging of live human cells. *Nature methods* **3,** 385–90. ISSN: 1548-7091 (2006).

84. Wang, M., Zhou, X., King, R. W. & Wong, S. T. C. Context based mixture model for cell phase identification in automated fluorescence microscopy. *BMC bioinformatics* **8,** 32. ISSN: 1471-2105 (2007).

85. Harder, N. *et al.* Automatic analysis of dividing cells in live cell movies to detect mitotic delays and correlate phenotypes in time. *Genome research* **19,** 2113–2124 (2009).

86. Snijder, B. *et al.* Single-cell analysis of population context advances RNAi screening at multiple levels. *Molecular systems biology* **8,** 579. ISSN: 1744-4292 (2012).

87. Huh, S., Ker, D. F. E., Su, H. & Kanade, T. Apoptosis detection for adherent cell populations in time-lapse phase-contrast microscopy images. *Med. Image Comput. Comput. Assist. Interv.* **15,** 331–339 (Jan. 2012).

88. Sadanandan, S. K., Ranefall, P., Le Guyader, S. & Wählby, C. Automated Training of Deep Convolutional Neural Networks for Cell Segmentation. *Scientific Reports.* ISSN: 20452322. doi:10.1038/s41598-017-07599-6 (2017).

89. Kandaswamy, C., Silva, L. M., Alexandre, L. A. & Santos, J. M. High-Content Analysis of Breast Cancer Using Single-Cell Deep Transfer Learning. *Journal of biomolecular screening* **21,** 252–9. ISSN: 1552-454X (2016).

90. Pawlowski, N., Caicedo, J. C., Singh, S., Carpenter, A. E. & Storkey, A. Automating Morphological Profiling with Generic Deep Convolutional Networks. *bioRxiv,* 4–8 (2016).

91. Warchal, S. J., Dawson, J. C. & Carragher, N. O. Evaluation of Machine Learning Classifiers to Predict Compound Mechanism of Action When Transferred across Distinct Cell Lines. *SLAS Discovery.* ISSN: 24725560. doi:10.1177/2472555218820805 (2019).

92. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **10008,** 6. ISSN: 1742-5468 (2008).

93. Kohonen, T. & Honkela, T. Kohonen network. *Scholarpedia* **2,** 1568 (2007).

94. Ng, A. Y. J. *et al.* A Cell Profiling Framework for Modeling Drug Responses from HCS Imaging. *Journal of Biomolecular Screening* **15,** 858–868. ISSN: 1087-0571 (2010).

95. Ge, Y. & Sealfon, S. C. flowPeaks: a fast unsupervised clustering for flow cytometry data via K-means and density peak finding. *Bioinformatics (Oxford, England)* **28,** 2052–8. ISSN: 1367-4811 (Aug. 2012).

96. *Hierarchical Clustering* scikit - learn . org / stable / modules / clustering . html # hierarchical-clustering (2019).

97. Bakal, C., Aach, J., Church, G. & Perrimon, N. Quantitative Morphologicla Signatures Define Local Signaling Networks Regulating Cell Morphology. *Science* **316,** 1753–1756. ISSN: 0036-8075 (2007).

98. Neumann, B. *et al.* Phenotypic profiling of the human genome by time-lapse microscopy reveals cell division genes. *Nature* **464,** 721–727. ISSN: 00280836 (2010).

99. Gustafsdottir, S. M. *et al.* Multiplex Cytological Profiling Assay to Measure Diverse Cellular States. **8,** 1–7 (2013).

100. Kang, J. *et al.* Improving drug discovery with high-content phenotypic screens by systematic selection of reporter cell lines. *Nat Biotech* **34,** 70–77. ISSN: 1087-0156 (2016).

101. Ester, M., Kriegel, H.-P., Sander, J., Xu, X., *et al. A density-based algorithm for discovering clusters in large spatial databases with noise.* in *Kdd* **96** (1996), 226–231.

102. *DBSCAN* scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html (2019).

103. Lutins, E. *DBSCAN: What is it? When to Use it? How to use it.* %7Bmedium.com/@elutins/dbscan-what-is-it-when-to-use-it-how-to-use-it-8bd506293818%7D (2019).

104. Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing* **17,** 395–416 (2007).

105. *ENS Paris-Saclay. M2 MVA Mathématiques / Vision / Apprentissage* math.ens-paris-saclay.fr/version-francaise/formations/master-mva/ (2019).

106. Rohban, M. H., Singh, S. & Carpenter, A. E. Capturing single-cell heterogeneity via data fusion improves image-based profiling. *bioRxiv Bioinformatics.* doi:10.1101/328542. http://biorxiv.org/cgi/content/short/328542v1 (2018).

107. Yin, Z. *et al.* Using iterative cluster merging with improved gap statistics to perform online phenotype discovery in the context of high-throughput RNAi screens. *BMC bioinformatics* **9,** 264. ISSN: 1471-2105 (2008).

108. Yin, Z. *et al.* How cells explore shape space: A quantitative statistical perspective of cellular morphogenesis. *BioEssays* **36,** 1195–1203. ISSN: 15211878 (2014).

109. *Flow Cytometry Powers High-Throughput Screening Advances* https://www.slas.org/eln/flow-cytometry-powers-high-throughput-screening-advances/ (2019).

110. Marvin, J. *Sorting at the Flow Cytometry Facility* utahflowcytometry . files . wordpress.com/2012/09/sorting-guidelines-2012.pdf (2019).

111. Adan, A., Alizada, G., Kiraz, Y., Baran, Y. & Nalbant, A. Flow cytometry: basic principles and applications. *Critical reviews in biotechnology* **37,** 163–176 (2017).

112. Naim, I. *et al.* SWIFT—scalable clustering for automated identification of rare cell populations in large, high-dimensional flow cytometry datasets, Part 1: Algorithm design. *Cytometry Part A* **85,** 408–421 (2014).

113. Sörensen, T., Baumgart, S., Durek, P., Grützkau, A. & Häupl, T. immunoClust—An automated analysis pipeline for the identification of immunophenotypic signatures in high-dimensional cytometric datasets. *Cytometry Part A* **87,** 603–615 (2015).

114. Lo, K., Hahne, F., Brinkman, R. R. & Gottardo, R. flowClust: a Bioconductor package for automated gating of flow cytometry data. *BMC bioinformatics* **10,** 145 (2009).

115. Zare, H., Shooshtari, P., Gupta, A. & Brinkman, R. R. Data reduction for spectral clustering to analyze high throughput flow cytometry data. *BMC bioinformatics* **11,** 403 (2010).

116. Levine, J. H. *et al.* Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. *Cell* **162,** 184–97. ISSN: 1097-4172 (July 2015).

117. Qiu, P. *et al.* Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nature biotechnology* **29,** 886 (2011).

118. Tibshirani, R., Walther, G. & Hastie, T. Estimating the number of clusters in a data set via the gap statistic. **63,** 411–423. ISSN: 1369-7412 (2001).

119. Domingos, P. A few useful things to know about machine learning. *Communications of the ACM* **55,** 78. ISSN: 00010782 (2012).

120. Wang, J. *Geometric structure of high-dimensional data and dimensionality reduction* (Springer, 2012).

121. Radovanović, M., Nanopoulos, A. & Ivanović, M. Hubs in Space : Popular Nearest Neighbors in High-Dimensional Data. *Journal of Machine Learning Research* **11,** 2487–2531. ISSN: 15324435 (2010).

122. Aggarwal, C. C., Hinneburg, A. & Keim, D. A. On the surprising behavior of distance metrics in high dimensional space. *Database Theory – ICDT 2001,* 420–434. ISSN: 0956-7925 (2001).

123. Houle, M. E., Kriegel, H.-p. & Kröger, P. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality ? Outline The Curse of Dimensionality Shared-Neighbor Distances Experimental Set-Up Observations, 482–500 (2010).

124. Zimek, A., Schubert, E. & Kriegel, H.-P. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* **5,** 363–387 (2012).

125. Dy, J. G. & Brodley, C. E. Feature Selection for Unsupervised Learning. *Journal of Machine Learning Research* **5,** 845–889. ISSN: 15337928 (2004).

126. Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. Feature selection for high-dimensional data. *Progress in Artificial Intelligence* **5,** 65–75 (2016).

127. Hall, M. A. Correlation-based feature selection of discrete and numeric class machine learning (2000).

128. Gutierrez-Osuna, R. *Introduction to Pattern Analysis. Lecture 11: Sequential Feature Selection* http://www.facweb.iitkgp.ac.in/~sudeshna/courses/ML06/featsel.pdf (2019).

129. Witten, I. H., Frank, E. & Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques* (Elsevier, 2011).

130. Dash, M. & Liu, H. Handling large unsupervised data via dimensionality reduction. *Proceedings of 1999 SIGMOD Research Issues in Data Mining and Knowledge Discovery (DMKD-99) Workshop,* 1–5 (1999).

131. Mitra, P., Murthy, C. & Pal, S. K. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence* **24,** 301–312 (2002).

132. Liu, H., Yu, L., Member, S. S., Yu, L. & Member, S. S. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on* **17,** 491–502. ISSN: 1041-4347 (2005).

133. Guyon, I. & Elisseeff, A. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research (JMLR)* **3,** 1157–1182. ISSN: 00032670 (2003).

134. Reisen, F., Zhang, X., Gabriel, D. & Selzer, P. Benchmarking of multivariate similarity measures for high-content screening fingerprints in phenotypic drug discovery. *Journal of biomolecular screening* **18,** 1284–97. ISSN: 1552-454X (2013).

135. Young, D. W. *et al.* Integrating high-content screening and ligand-target prediction to identify mechanism of action. *Nature chemical biology* **4,** 59–68. ISSN: 1552-4450 (2008).

136. Levina, E. & Bickel, P. J. *Maximum likelihood estimation of intrinsic dimension* in *Advances in neural information processing systems* (2005), 777–784.

137. Camastra, F. Data dimensionality estimation methods: A survey. *Pattern Recognition* **36,** 2945–2954. ISSN: 00313203 (2003).

138. Tenenbaum, J. B., De Silva, V. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *science* **290,** 2319–2323 (2000).

139. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science* **290,** 2323–2326 (2000).

140. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research* **9,** 2579–2605 (2008).

141. Vathy-fogarassy, A., Kiss, A. & Abonyi, J. Topology Representing Network Map – A New Tool for Visualization of High-Dimensional Data, 61–84.

142. Camastra, F. & Vinciarelli, A. Estimating the intrinsic dimension of data with a fractal-based method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24,** 1404–1407. ISSN: 01628828 (2002).

143. Verveer, P. J., Verveer, P. J. & Duin, R. P. W. An Evaluation of Intrinsic Dimensionality Estimators. **17,** 81–86. ISSN: 01628828 (1995).

144. Costa, J. A. & Hero, A. O. Geodesic entropic graphs for dimension and entropy estimation in Manifold learning. *IEEE Transactions on Signal Processing* **52,** 2210–2221. ISSN: 1053587X (2004).

145. Crosetto, N., Bienko, M. & Oudenaarden, A. V. Spatially resolved transcriptomics and beyond. **16** (2015).

146. Hansson, K., Jafari-Mamaghani, M. & Krieger, P. RipleyGUI: software for analyzing spatial patterns in 3D cell distributions. *Frontiers in Neuroinformatics* **7,** 1–9. ISSN: 1662-5196 (2013).

147. Li, Y., Majarian, T. D., Naik, A. W., Johnson, G. R. & Murphy, R. F. Point Process Models for Localization and Interdependence of Punctate Cellular Structures. *Cytometry Part A* **89A,** 633–643 (2016).

148. Arpon, J. *Statistical analysis and modeling of nuclear architecture in Arabidopsis Thaliana.* PhD thesis (Université Paris-Saclay, 2016).

149. Alexandrov, T. & Bartels, A. Testing for presence of known and unknown molecules in imaging mass spectrometry. *Bioinformatics* **29,** 2335–2342. ISSN: 13674803 (2013).

150. Zhu, Q., Shah, S., Dries, R., Cai, L. & Yuan, G.-c. Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data. *Nature Publishing Group.* ISSN: 1087-0156. doi:10.1038/nbt.4260. http://dx.doi.org/10.1038/nbt.4260 (2018).

151. Gavrikov, V. & Stoyan, D. The use of marked point processes in ecological and environmental forest studies. *Environmental and Ecological Statistics* **2,** 331–344. ISSN: 13528505 (1995).

152. Penttinen, A. & Stoyan, D. Recent applications of point process methods in forestry statistics. *Statistical Science* **15,** 61–78. ISSN: 0883-4237 (2000).

153. Fauchald, P., Erikstad, K. E. & Skarsfjord, H. Scale-dependent predator–prey interactions: the hierarchical spatial distribution of seabirds and prey. *Ecology* **81,** 773–783 (2000).

154. Legendre, P. & Legendre, L. *Structure functions* 2nd Englis. **2,** 712–738. doi:10.1093/oxfordjournals.bjc.a046803 (Elsevier Science BV, Amsterdam, 1998).

155. Moran, P. A. Notes on continuous stochastic phenomena. *Biometrika* **37,** 17–23 (1950).

156. Overmars, K. P., De Koning, G. H. & Veldkamp, A. Spatial autocorrelation in multi-scale land use models. *Ecological Modelling* **164,** 257–270. ISSN: 03043800 (2003).

157. Dixon, P. M. Ripley's K function. *Encyclopedia of Environmetrics* **3,** 1796–1803 (2002).

158. Szmyt, J. Spatial statistics in ecological analysis: From indices to functions. *Silva Fennica* **48.** ISSN: 22424075. doi:10.14214/sf.1008 (2014).

159. Suzuki, S. N., Kachi, N. & Suzuki, J. I. Development of a local size hierarchy causes regular spacing of trees in an even-aged Abies forest: Analyses using spatial autocorrelation and the mark correlation function. *Annals of Botany* **102,** 435–441. ISSN: 03057364 (2008).

160. McPherson, M., Smith-Lovin, L. & Cook, J. M. Birds of a feather: Homophily in social networks. *Annual review of sociology* **27,** 415–444 (2001).

161. Newman, M. E. Mixing patterns in networks. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics* **67,** 13. ISSN: 1063651X (2003).

162. Easley, D., Kleinberg, J., *et al. Networks, crowds, and markets* (Cambridge university press Cambridge, 2010).

163. Hagberg, A., Swart, P. & S Chult, D. *Exploring network structure, dynamics, and function using NetworkX* tech. rep. (Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008).

164. *Spatial Regession in R 1: The Four Simplest Models* youtu.be/b3HtV2Mhmvk (2019).

165. Mantel, N. The detection of disease clustering and a generalized regression approach. *Cancer research* **27,** 209–220 (1967).

166. Dray, S. & Jombart, T. Revisiting Guerry's data: Introducing spatial constraints in multivariate analysis. *Annals of Applied Statistics* **5,** 2278–2299. ISSN: 19326157 (2011).

167. Fuller, M. M., Wagner, A. & Enquist, B. J. Using Network Analysis To Characterize Forest Structure. *Natural Resource Modeling* **21,** 225–247 (2008).

168. (ESRI), E. S. R. I. *ArcGIS* 2012.

169. Rey, S. J. & Anselin, L. PySAL: A Python library of spatial analytical methods, 175–193 (2010).

170. Baddeley, A., Rubak, E. & Turner, R. *Spatial point patterns: methodology and applications with R* (Chapman and Hall/CRC, 2015).

171. Murphy, R. F. CellOrganizer: Image-derived models of subcellular organization and protein distribution. **110,** 179–193 (2012).

172. Edsgärd, D., Johnsson, P. & Sandberg, R. Identification of spatial expression trends in single-cell gene expression data. **15.** doi:10.1038/nmeth.4634 (2018).

173. Svensson, V., Teichmann, S. A. & Stegle, O. SpatialDE: identification of spatially variable genes. **15.** doi:10.1038/nmeth.4636 (2018).

174. Feuillet, T. *et al.* Spatial heterogeneity of the relationships between environmental characteristics and active commuting: Towards a locally varying social ecological model. *International Journal of Health Geographics* **14,** 1–14. ISSN: 1476072X (2015).

175. Gunduz, C., Ener, B. Y. & Gultekin, S. H. The cell-graphs of cancer. *BIOINFORMATICS* **00,** 1–7 (2004).

176. Yener, B. Cell-Graphs: Image-Driven Modeling of Structure- Function Relationship. *Communication of the ACM* **60,** 74–84 (2017).

177. Pavlopoulos, G. A. *et al.* Using graph theory to analyze biological networks. *BioData Mining* **4,** 10. ISSN: 1756-0381 (2011).

178. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *science* **286,** 509–512 (1999).

179. Albert, R. Scale-free networks in cell biology. *Journal of Cell Science* **118,** 4947–4957. ISSN: 0021-9533 (2005).

180. Samato, N. F. *Practical Graph Mining with R - Graph Classification* www.csc2.ncsu.edu/faculty/nfsamato/practical-graph-mining-with-R/slides/pdf/Classification.pdf (2019).

181. Riazi, S. Graph Representation Learning and Graph Classification (2017).

182. Zhu, X. & Ghahramani, Z. Learning from labeled and unlabeled data with label propagation (2002).

183. Valko, M. *Graphs in machine learning* math.ens-paris-saclay.fr/version-francaise/formations/master-mva/contenus-/graphs-in-machine-learning-267194.kjsp (2019).

184. Cook, S. A. *The complexity of theorem-proving procedures* in *Proceedings of the third annual ACM symposium on Theory of computing* (1971), 151–158.

185. Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **35,** 1798–1828 (2013).

186. Ozdemir, E. & Gunduz-demir, C. A Hybrid Classi fi cation Model for Digital Pathology Using Structural and Statistical Pattern Recognition. **32,** 474–483 (2013).

187. Haussler, D. Convolution kernels on discrete structures (1999).

188. Borgwardt, K. & Yan, X. *Graph Mining and Graph Kernels. Part II: Graph Kernels* sites.cs.ucsb.edu/~xyan/tutorial/KDD08_graph_partII.pdf (2019).

189. Borgwardt, K. M., Ong, C. S., Vishwanathan, S. V. N., Smola, A. J. & Kriegel, H.-p. Protein function prediction via graph kernels. **21,** 47–56 (2005).

190. Vega, M. A. *Graph kernels and applications in chemoinformatics* PhD thesis (Utah State University, 2011).

191. Ralaivola, L., Swamidass, S. J., Saigo, H. & Baldi, P. Graph Kernels for Chemical Informatics.

192. Shervashidze, N., Schweitzer, P., Leeuwen, E. J. v., Mehlhorn, K. & Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* **12,** 2539–2561 (2011).

193. Sugiyama, M., Ghisu, M. E., Llinares-lo, F. & Borgwardt, K. graphkernels: R and Python packages for graph comparison, 1–3 (2017).

194. Tixier, A. J.-P., Nikolentzos, G., Meladianos, P. & Vazirgiannis, M. Classifying Graphs as Images with Convolutional Neural Networks, 1–13 (2017).

195. Vishwanathan, S. V. N., Borgwardt, K. M., Kondor, I. R. & Schraudolph, N. N. Graph Kernels. **9,** 1–41 (2008).

196. Li, C., Guo, X. & Mei, Q. Deepgraph: Graph structure predicts network growth. *arXiv preprint arXiv:1610.06251* (2016).

197. Niepert, M., Ahmed, M. & Kutzkov, K. Learning Convolutional Neural Networks for Graphs. **1.** arXiv: arXiv:1605.05273v4 (2015).

198. Kipf, T. N. & Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907* (2016).

199. *BioPhenics Institut Curie* biophenics.net/ (2019).

200. Rose, F. *et al.* Compound Functional Prediction Using Multiple Unrelated Morphological Profiling Assays. *SLAS Technology* **23,** 243–251. ISSN: 24726311 (Nov. 2018).

201. *Human MCF7 cells – compound-profiling experiment* data.broadinstitute.org/bbbc/BBBC021/ (2019).

202. Caie, P. D. *et al.* High-Content Phenotypic Profiling of Drug Response Signatures across Distinct Cancer Cells. *Molecular cancer therapeutics* **9,** 1913–26. ISSN: 1538-8514 (June 2010).

203. Kraus, O. Z., Ba, J. L. & Frey, B. J. Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics* **32,** i52–i59. ISSN: 14602059 (2016).

204. Toth, T. *et al.* Environmental properties of cells improve machine learning-based phenotype recognition accuracy. *Scientific Reports* **8,** 1–9. ISSN: 20452322 (2018).

205. Piccinini, F. *et al.* Advanced Cell Classifier: User-Friendly Machine-Learning-Based Software for Discovering Phenotypes in High-Content Imaging Data. *Cell Systems* **4,** 651–655.e5. ISSN: 2405-4712 (2017).

206. Zhang, W., Wang, J., Jin, D., Oreopoulos, L. & Zhang, Z. *A deterministic self-organizing map approach and its application on satellite data based cloud type classification* in *2018 IEEE International Conference on Big Data (Big Data)* (2018), 2027–2034.

207. Arthur, D. & Vassilvitskii, S. k-means++: The advantages of careful seeding, 1027–1035 (2007).

208. Kraus, O. Z. *et al.* Automated analysis of high-content microscopy data with deep learning, 1–15 (2017).

209. Shervashidze, N., Mehlhorn, K., Lafayette, W., Petri, T. H. & Borgwardt, K. M. Efficient graphlet kernels for large graph comparison. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)* **5,** 488–495 (2009).

210. *Alexandrov Team. Spatial Metabolomics* www.embl.de/research/units/scb/alexandrov/ (2019).

211. *Metaspace, Molecular Annotation Engine* project.metaspace2020.eu/ (2019).

212. *Division of Chronic Inflammation and Cancer* https://www.dkfz.de/en/chronische-entzuendung-und-krebs/index.php (2019).

213. Willebrords, J. *et al.* Strategies, models and biomarkers in experimental non-alcoholic fatty liver disease research. *Progress in lipid research* **59,** 106–125 (2015).

214. Czamara, K. *et al.* Unsaturated lipid bodies as a hallmark of inflammation studied by Raman 2D and 3D microscopy. *Scientific reports* **7,** 40889 (2017).

215. Diggle, P. J. *et al. Statistical analysis of spatial point patterns.* (Academic press, 1983).

216. *Human Cell Atlas* (2019).

217. Boyd, J. C., Pinheiro, A., Del Nery, E., Reyal, F. & Walter, T. Domain-invariant features for mechanism of action prediction in a multi-cell-line drug screen. *bioRxiv.* doi:10.1101/656025. eprint: https://www.biorxiv.org/content/early/2019/06/03/656025.full.pdf. https://www.biorxiv.org/content/early/2019/06/03/656025 (2019).

218. *chEMBL* https://www.ebi.ac.uk/chembl/ (2019).

219. *PubChem* https://pubchem.ncbi.nlm.nih.gov/ (2019).

220. *Pharos* https://www.pharosproject.net/ (2019).

221. Gillet, J. P., Varma, S. & Gottesman, M. M. The clinical relevance of cancer cell lines. *Journal of the National Cancer Institute* **105,** 452–458. ISSN: 00278874 (2013).

222. Hynds, R. E., Vladimirou, E. & Janes, S. M. The secret lives of cancer cell lines. *Disease Models & Mechanisms* **11,** dmm037366. ISSN: 1754-8403 (2018).

223. Ben-David, U. *et al.* Patient-derived xenografts undergo mouse-specific tumor evolution. *Nature genetics* **49,** 1567 (2017).

224. *Mimetas, the organ-on-chip company* https://mimetas.com/page/technology (2019).

225. Brassard, J. A. & Lutolf, M. P. Engineering Stem Cell Self-organization to Build Better Organoids. *Cell Stem Cell* **24,** 860–876 (2019).

226. *FlowCAP - Flow Cytometry: Critical Assessment of Population Identification Methods* http://flowcap.flowsite.org/ (2017).

227. Brasko, C. *et al.* Intelligent image-based in situ single-cell isolation. *Nature communications* **9,** 226 (2018).

228. Kester, L. & van Oudenaarden, A. Single-cell transcriptomics meets lineage tracing. *Cell Stem Cell* **23,** 166–179 (2018).

229. Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology* **32,** 381 (2014).

230. Gut, G., Tadmor, M. D., Pe'er, D., Pelkmans, L. & Liberali, P. Trajectories of cell-cycle progression from fixed cell populations. *Nature Methods* **12,** 951–954. ISSN: 15487105 (2015).

231. Evans, L., Sailem, H., Vargas, P. P. & Bakal, C. Inferring signalling networks from images. *Journal of Microscopy* **252,** 1–7. ISSN: 00222720 (2013).

232. Reisen, F. *et al.* Linking Phenotypes and Modes of Action Through High-Content Screen Fingerprints. *Assay and drug development technologies* **13,** 415–427. ISSN: 1557-8127 (2015).

233. Rohban, M. H., Abbasi, H. S., Singh, S. & Carpenter, A. E. Capturing single-cell heterogeneity via data fusion improves image-based profiling. *Nature communications* **10,** 2082 (2019).

234. Serra, D. *et al.* Self-organization and symmetry breaking in intestinal organoid development. *Nature,* 1 (2019).

RÉSUMÉ

La robotique et l'automatisation des microscopes ont ouvert la voie aux cribles cellulaires à haut contenu : des marqueurs fluorescents ciblant l'ADN ou d'autres composants sont utilisés pour imager des centaines de milliers de cellules dans différentes conditions. Il a été montré que les cribles cellulaires sont efficaces pour découvrir des médicaments de nouvelles classes thérapeutiques, cad ceux qui agissent sur une nouvelle cible. Les cribles permettent d'identifier des composés prometteurs et de les caractériser en leur associant des annotations fonctionnelles, comme leur cible moléculaire ou leur mécanisme d'action (MOA). J'ai étudié l'hétérogénéité des réponses cellulaires à différents niveaux et comment cette hétérogénéité phénotypique peut être exploitée pour mieux caractériser les composés. Au premier niveau, j'ai étudié l'hétérogénéité entre patients. Nous avons montré qu'utiliser différentes lignées cellulaires dérivées de patients augmente la probabilité de prédire la cible moléculaire du composé testé. Le second niveau correspond à la diversité des réponses cellulaires de la même lignée cellulaire soumise au même traitement. Des méthodes de clustering appropriées peuvent être utilisées pour clarifier cette complexité et pour grouper les cellules en sous-populations. Les proportions de chaque sous-population par traitement permettent de prédire le bon MOA. Le troisième niveau regarde comment les sous-populations cellulaires sont organisées spatialement. J'ai trouvé que les cellules voisines s'influencent les unes les autres et affichent un phénotype similaire plus fréquemment qu'attendu par chance. Ces résultats obtenus sur une centaine de traitements montrent que des cellules génétiquement identiques ne sont pas identiques et indépendantes mais sont à l'origine d'une hétérogénéité spatiale par le lignage cellulaire et les interactions. En utilisant l'information spatiale ainsi que l'hétérogénéité phénotypique, les méthodes à noyaux de graphes améliorent la classification en MOA sous certaines conditions. Parallèlement, comme l'analyse spatiale peut s'appliquer à n'importe quelle image de microscopie, j'ai développé une librairie d'analyse *Python*, *PySpacell*, pour étudier l'aléatoire spatial de marqueurs quantitatifs et qualitatifs.

MOTS CLÉS

Criblage à haut contenu, analyse d'images, phénotypage cellulaire, statistiques spatiales

ABSTRACT

Robotics and automated fluorescence microscopes have promoted high-content cell-based screenings: fluorescent probes targeting DNA or other major components are used to image hundreds of thousands of cells under many different conditions. Cell-based assays have proven to be efficient at discovering first-in-class therapeutic drugs, i.e. drugs acting on a new target. They allow to detect promising molecules and to profile them, by associating functional annotations to them, like their molecular target or mechanism of action (MOA). I studied heterogeneity of cell responses at different levels and how this phenotypic heterogeneity can be leveraged to better profile drugs. The first level is about studying heterogeneity between patients. We showed that using different patient-derived cell lines increases the chance of predicting the correct molecular target of the tested drug. The second level corresponds to the diversity of cell responses within the same cell line under the same treatment. Appropriate clustering approaches can be used to unravel this complexity and group cells into subpopulations. The proportions of each subpopulation per treatment allow to predict the correct MOA. The third level looks at how the cell subpopulations are spatially organized. I found that neighboring cells influence each others, and display a similar phenotype more frequently than expected at random. These results assessed across a hundred of treatments, show that even genetically identical cells are not all alike and independent, but create spatial heterogeneity via cell lineage and interaction. Using spatial information as well as phenotypic heterogeneity with graph kernel methods improves the MOA classification under some conditions. Alongside, as spatial analysis could be applied on any cell microscopy image, I developed a *Python* analysis package, *PySpacell*, to study spatial randomness from quantitative and qualitative cell markers.

KEYWORDS

High content screening, image analysis, cell phenotyping, spatial statistics