A generic methodological framework for studying single cell motility in high-throughput time-lapse data

Alice Schoenauer Sebag1,2,3,4,5,6, Sandra Plancade7, Céline Raulet-Tomkiewicz4,5, Robert Barouki4,5, Jean-Philippe Vert1,2,3 and Thomas Walter1,2,3*

1MINES ParisTech, PSL-Research University, CBIO-Centre for Computational Biology, Fontainebleau, 2Institut Curie, Paris, 3INSERM U900, Paris, 4Université Paris Descartes, Paris, 5INSERM UMR-S 1124, Paris, 6Agro ParisTech, Paris and 7Mathématiques et Informatique Appliquées, INRA, Jouy-en-Josas, France

*To whom correspondence should be addressed.

Abstract

Motivation: Motility is a fundamental cellular attribute, which plays a major part in processes ranging from embryonic development to metastasis. Traditionally, single cell motility is often studied by live cell imaging. Yet, such studies were so far limited to low throughput. To systematically study cell motility at a large scale, we need robust methods to quantify cell trajectories in live cell imaging data.

Results: The primary contribution of this article is to present Motility study Integrated Workflow (MotIW), a generic workflow for the study of single cell motility in high-throughput time-lapse screening data. It is composed of cell tracking, cell trajectory mapping to an original feature space and hit detection according to a new statistical procedure. We show that this workflow is scalable and demonstrates its power by application to simulated data, as well as large-scale live cell imaging data. This application enables the identification of an ontology of cell motility patterns in a fully unsupervised manner.

Availability and implementation: Python code and examples are available online (http://cbio.ensmp.fr/~aschoenauer/motiw.html)

Contact: thomas.walter@mines-paristech.fr

Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

High-throughput (HT) microscopy and high content screening (HCS) are state-of-the-art approaches to investigate many aspects of cellular organization and function, such as protein localization and function, such as protein localization (Glory and Murphy, 2007), spatial transcriptomics (Battich et al., 2013) or drug screening (Perlman et al., 2004).

In particular, when combined with a loss-of-function strategy, these approaches are now widely used to study the molecular basis of biological processes by monitoring the phenotypic consequences of downregulation or overexpression of genes of interest (Pepperkok and Ellenberg, 2006). When performed at a large and ideally genome-wide scale, such screening approaches have become indispensable tools for functional genomics: they have the potential to provide us with a close to complete picture of the proteins involved in the process under study.

Indeed, many large-scale phenotypic screens have been published previously, shedding light on the regulation of such diverse cellular processes as protein secretion (Simpson et al., 2012), endocytosis (Collinet et al., 2010) or cell division (Neumann et al., 2010). Such screens do not only provide lists of candidate genes for further follow-up studies and for computational modeling approaches, they also generate large image databases. Those can turn out to be a precious scientific resource—as a collection of experimental data for punctual queries or as a basis for systematic and potentially integrative computational analysis. Although it has always been a strength in bioinformatics to rely on rich publicly available data sources, the use and re-use of image data is not straightforward. On the one hand, this is due to the lack of standardized data formats (Sommer et al., 2013) and ontologies (Hoehndorf et al., 2012). On the other hand, the computational tools which are necessary to perform such
analyses are often not available. As a consequence, studies based on
remining existing phenotypic screens have only recently started to
appear (Ostaszewski et al., 2012; Pau et al., 2013; Suratanee et al.,
2014). We hypothesize that remining these rich phenotypic re-
resources can help to increase our understanding of many basic cellular
functions, such as cell motility.

Cell motility plays a key role in many physiological processes
including embryonic development or immune response (Friedl and
Weigelin, 2008), and is also involved in pathological processes such as
fibrosis and metastasis. The latter is dependent on the ability of
cancer cells to migrate, both as single cells and collectively
(Decaestecker et al., 2007; Yilmaz and Christofori, 2010).

Many in vitro assays have been specifically designed to study cell
motility (Decaestecker et al., 2007; Kramer et al., 2013). Examples
include semi-automated analysis of single cell traces on bead-coated
layers (Naffar-Abu-Amara et al., 2008) and wound healing assays
measuring collective cell migration (Simpson et al., 2008).

However, single cell motility studies on live cell imaging data
have so far been limited to low-to-medium (Lara et al., 2011)
throughput. There are two bottlenecks that currently explain the
non-existence of such studies in an HT setup: acquiring large-scale data
is expensive, and the relevant computational tools are often either
non-existent or not easily scalable.

The contribution of this article is to present MotIW (Motility
study Integrated Workflow), and its application to motility gene dis-
covery. A generic methodological workflow, MotIW enables to
quantitatively study cell motility at single cell resolution in HT time-
lapse data in an unsupervised way. It consists of cell tracking, cell
trajectory mapping to an original feature space and outlier experi-
ment detection according to a new statistical procedure (Fig. 1). We
show the power of our method by applying MotIW to simulated
data, which allows us to estimate recall and precision to be expected
on real data. We then apply this workflow to a previously published
genome-wide screen by RNA interference (RNAi) and live cell imaging,
the Mitocheck dataset. Analysis of the screening data reveals the
existence of a cell trajectory ontology in the dataset. Without any
prior assumption on cell motion, we are able to identify eight
types of cell trajectories.

The remainder of this article is organized as follows: after a short
description of the data and the implementation details in Section 2,
we detail MotIW in Sections 3.1–3.3, as well as its application to
simulated data in Section 3.4 and the Mitocheck screen in Section 3.5.
Section 4 briefly discusses broader perspectives of this workflow.

2 Materials and methods

2.1 A genome-wide time-resolved dataset

We used a previously published genome-wide dataset of time-
resolved records of cellular phenotype responses to gene silencing,
which were generated for virtually all protein-coding genes
(Neumann et al., 2010). It is publicly available at mitocheck.org.

For this, arrays of transfection cocktails containing small inter-
ferring RNAs (siRNAs) were spotted directly in live cell-imaging
chambers in a 384 format. HeLa cells stably expressing the core his-
tone 2B tagged with green fluorescent protein (GFP) were seeded
on top of the arrays, and imaged 18 h after the transfection for 48 h
with a time lapse of 30 min (Plan10x, NA 0.4; Olympus). Each
microarray contained 8 negative controls (scrambled: not targeting
any gene) and 12 positive controls showing different phenotypes. A
total of 22 612 protein-coding genes have been targeted by at least 2
siRNAs each, in total 51 767 siRNAs. For each siRNA, there are
data from at least 3 technical replicates, which created 182 191
quality controlled time-lapse experiments in total. Because of up-
dates in the genome annotation, some reagents could not be mapped
to the current ENSEMBL version. In total, the dataset contains data
for 17 816 protein-coding genes in 144 909 quality controlled time-
lapse experiments.

2.2 Software

We use CellCognition (Held et al., 2010) (cellcognition.org) for
segmentation and feature extraction and CPLEX (http://www-01.ibm.
com/software/commerce/optimization/cplex-optimizer/) for opti-
mization in the tracking procedure. To store, manage and access
the screening data, we use a previously published data format CellH5
(Sommer et al., 2013). All scripts are written in the programming
language Python 2.7 using SciPy (Jones et al. 2001) and NumPy,
and all plots were generated by matplotlib (Hunter 2007).

3 Results

In this section, we present MotIW, our workflow for the automatic
and quantitative analysis of single cell motility in video sets from
time-lapse microscopy-based screens. Figure 1 summarizes its dif-
ferent steps. Briefly, for each video nuclei are segmented and features
are extracted as published previously (Held et al., 2010; Walter
et al., 2010). Cells are tracked using a new machine learning-based
tracking procedure, described in Section 3.1. The trajectories are
then mapped to a feature space described in Section 3.2. Presented
in Section 3.3, an original statistical procedure then enables the de-
tection of experiments in which single cell motility is significantly
different than that in control movies. Finally, Section 3.4 describes
the simulation of trajectories which allows us to validate the per-
formance of the workflow.

3.1 Cell tracking

There exist two main approaches to cell tracking. The deformable
model approach relies on identifying and modeling objects on the
first frame, and linking them to objects in consecutive frames by
updating the models (Zimmer et al., 2002). On the other hand, the
object association approach associates preidentified objects in con-
secutive frames (Lou and Hamprecht, 2011).
Different datasets usually require different object identification approaches. For the sake of modularity, we therefore prefer to keep segmentation and tracking steps independent.

Cell tracking faces several challenges in videos from high content screens like Mitocheck, including high population density in each picture, high phenotypic inter-cell variability and possibly low time resolution between successive images (30 min). Furthermore, the algorithm has to handle apparitions, disappearances, divisions and fusions (this event results from occlusion or segmentation errors). Finally, to be applicable in a screening context, we cannot a priori model cell motion, as such hypotheses are bound to break in the presence of phenotypes. Indeed, the impact of chemical exposure on cell motion is not known. To also avoid dependence on manual parameter tuning, we have extended a non-parametric structured learning approach from Lou and Hamprecht (2011).

We first characterize each cell in each image by a set of 230 object features including geometric, shape and texture features (Held et al., 2010; Walter et al., 2010). The goal of cell tracking in this approach is to match cells in successive images, by assigning them the most likely instant temporal behavior in the set \( E = \{ \text{move, appear, disappear, split in 2 or 3, merge at 2 or 3} \} \). All possible matches between cells in consecutive frames are exhaustively considered, subject to distance thresholding. Match features are the following:

- The absolute difference in object features if the event is move, split, merge, the object features otherwise
- The geometrical distance between object at time \( t \), \( \text{Obj}_{ij,t} \), and object at time \( t+1 \), \( \text{Obj}_{ij,t+1} \), if the event is move, split, merge, the minimal distance to the image border otherwise
- The maximal angle between \( \text{Obj}_{ij,t} \) and the elements of \( \text{Obj}_{ij,t+1} \), if the event is a split
- The angle between the main axis of \( \text{Obj}_{ij,t} \) and \( \text{Obj}_{ij,t+1} \) weighted by their average eccentricity, if the event is a move

The optimal object matching \( \hat{z}(t) \) comes down to bi-partite graph matching: it is solved by maximizing a likelihood function \( L \), which depends on the weights \( w \) of match features and the match features \( f_{ij} \), subject to the constraint that all objects are matched in both frames [cf Equation (1)].

\[
\hat{z}(t) = \arg\max_{z(t)} L(z(t); w) \tag{1}
\]

where

\[
L(z(t); w) = \sum_{i \in X_{ij}} w^* f_{ij}^* > z_{ij}^*(t)
\]

s.t. \( \forall i \sum_{\text{Obj}_{ij,t}} z_{ij}^*(t) = 1 \)

and \( \forall j \sum_{\text{Obj}_{ij,t+1}} z_{ij}^*(t) = 1 \)

The weights \( w \) are learned by a support vector machine using annotated trajectories, following the formulation of Lou and Hamprecht (2011). The likelihood maximization, an integer linear programming (ILP) problem, is solved by IBM Cplex.

The extension compared with Lou and Hamprecht (2011) lies in the choice of match features. We also implemented a more efficient computation of match hypotheses using kd-trees. Furthermore, we enabled the tracking model to learn from partial annotations of different experiments. (However, this is not learning from partial annotations in the sense of Lou and Hamprecht (2012). Indeed, in our implementation of Lou and Hamprecht (2011), the user chooses a subset of cells which has to be annotated on all movie frames. In Lou and Hamprecht (2012), the user can choose both a subset of movie frames and a subset of cells (she wishes to annotate on those frames). This permits the user to integrate examples from both control and non-control experiments in the training set, which is crucial to guarantee that the model can efficiently track cells in all conditions. We also added three object division and fusion to \( E \). This is important in a screening context, where aberrant cell divisions may occur. In the future, it could be interesting to couple object segmentation and tracking to correct for missing detections.

To validate MotIW’s cell tracking procedure, we compare it with CellCognition’s constrained nearest-neighbor (CNN) tracking algorithm, and with Jaqaman et al. (2008) as implemented in CellProfiler (Carpenter et al., 2006). The latter approach views tracking as a linear assignment problem (LAP) and uses user-defined costs for performing merges, splits, appearances and disappearances. We have chosen these two approaches for benchmarking, as they are available in popular High Content Screening software.

Our training set consists of \( \sim 32000 \) matches, among which 0.5% appear, 0.5% disappear, 1% merge and 2% split. They come from the Mitocheck dataset. Furthermore, they were taken from both control experiments and phenotypic experiments according to Neumann et al. (2010) in order to ensure that the algorithm also works in the presence of phenotypes. The training set was annotated using CellCognition’s CNN tracking algorithm followed by manual correction. As shown in Table 1, MotIW outperforms the other two methods as measured by the average accuracy on the five movement types. As can be seen in Figure 2, they show similar performances on move events and have therefore similar overall (pooled) accuracies. The contribution of the learning approach is most important for the other events, such as cell division, when object matching is less trivial.

### 3.2 Trajectory features

Once cell trajectories are captured, each trajectory is described by a set of 15 features. These features were partly taken from previous publications on quantitative motility analysis, partly newly designed.

Robust and precise features are needed to account for the partial stochasticity of cell migratory behavior. We use three types of features, as detailed in Table 2. Prior to that, trajectories resulting from object fusion and trajectories which are shorter than 10 frames are discarded. This trajectory quality control ensures that cell clusters are not considered, and increases the dataset robustness.

#### 3.2.1 Particle motion features

This group of features encompasses the diffusion coefficient and the movement type, which were in the first place used to study particle motion (see Ferrari et al., 2001; and one of its applications to single particle motion in Biology—Sbalzarini and Koumoutsakos, 2005).


Table 2. Cell trajectory features and their formulas

<table>
<thead>
<tr>
<th>Feature</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diffusion coefficient</td>
<td>According to Shalzarni and Koumoutsakos (2005)</td>
</tr>
<tr>
<td>Diffusion adequation</td>
<td>Correlation between MSD(t) and t</td>
</tr>
<tr>
<td>Movement type</td>
<td>According to Shalzarni and Koumoutsakos (2005)</td>
</tr>
<tr>
<td>Englobing ball number</td>
<td>See text</td>
</tr>
<tr>
<td>Track entropy</td>
<td>See text</td>
</tr>
<tr>
<td>Convex hull area</td>
<td>—</td>
</tr>
<tr>
<td>Effective path length</td>
<td>$L =</td>
</tr>
<tr>
<td>Effective speed</td>
<td>$L/\sqrt{T}$</td>
</tr>
<tr>
<td>Largest move</td>
<td>—</td>
</tr>
<tr>
<td>Straightness index</td>
<td>$\sqrt{TL/P}$</td>
</tr>
<tr>
<td>Track curvature</td>
<td>See text</td>
</tr>
<tr>
<td>Mean squared displacement (MSD)</td>
<td>$\frac{1}{T} \sum_{i=1}^{T}</td>
</tr>
<tr>
<td>Mean signed turning angle</td>
<td>$\arctan \left( \frac{\text{sum}_{(x_i, y_i)}</td>
</tr>
</tbody>
</table>

Note: Notations: $(m_i)_{i=1}^T$, time sequence of cell 2D positions; $T$, track time duration; $P$, total track length.

Let us note $<d^p>$ the moment of order $p$ of a particle or a group of particles. For large $t$, it is proportionate to $t^p$ for most dispersive processes (Ferrari et al., 2001). Assuming that $\gamma_p$ is proportionate to $p$ (i.e. the particle movement is strongly self-similar), the constant $\gamma = \gamma_p/p$ (hereafter the particle’s Movement type) quantifies how directed the particle motion is. If $\gamma$ is equal to 1, the movement is perfectly directed, whereas if $\gamma$ is equal to 0.5, it is perfectly diffusive. Between 0.5 and 1, the movement is super-diffusive, whereas below 0.5 it is called sub-diffusive.

Furthermore, assuming $\gamma = 0.5$, the constant linking $<d^2 >$ [i.e. the mean squared displacement (MSD)] and $t$ can be computed—it is the Diffusion coefficient. The Diffusion adequation is the correlation coefficient between $<d^2 >$ and $t$, hence measuring how well the diffusive model applies to the track at hand.

We have furthermore created two new features to characterize the alternance between periods of diffusive motion and periods of directed motion—the track entropy and the englobing ball number.

It has been observed that cell motion in 2D alternates between diffusive and directed motions (in the absence of any perturbation or chemical gradient). The feature track Entropy was designed to measure how the time sequence of 2D cell positions $m_i = (x_i, y_i)$ distributes in balls of radius $r$ (see also Figure 3). This feature is calculated according to the following procedure, for each track of time duration $T$:

1. $S = \{1, \ldots, T\}$
2. while $S \neq \{\}$:
   i. do $t^* \leftarrow \argmax_{t} \text{card}(B_t(t))$
   where $B_t(t) = \{i \mid ||m_i - m_t||_2 \leq r$ and $\min(||m_{i+1} - m_t||_2, ||m_{i-1} - m_t||_2) \leq r\}$
   ii. do $S \leftarrow S \setminus B_t(t^*)$
3. Compute the track Entropy according to the following formula:

$$\text{Entropy}_r = -\frac{1}{T} \sum_{B_t} \frac{\text{card}(B_t)}{T} \log \left( \frac{\text{card}(B_t)}{T} \right)$$

(2)

The track Entropy measures the entropy of the distribution of track positions in balls of radius $r$. To deal with cells whose trajectories are concentrated in space, but were not concentrated in time, the constraint is imposed that these balls shall contain only consecutive positions in time.

The englobing Ball number is the number of balls of radius $r$ that contain all track positions. It is normalized by the square root of $T$ to be independent of the track time length $T$.

Different radii may be relevant for different data (depending on, e.g. the experiment time lapse, the pixel size or the cell type). We chose to use two different radii, $r_1$ and $r_2$ with $r_1 < r_2$, to incorporate information about cell trajectories on two different time scales. $r_1$ and $r_2$ were manually chosen, such that for the Mitocheck dataset, the corresponding features are neither constant nor too correlated. They respectively correspond to ~2.5 and 12 μm. In the following, the features Entropy $i$ and Ball number $i$ correspond to radius $r_i$.

3.2.2 Other global features

This group of features encompasses further global descriptors of the cell trajectory, such as the track Convex hull area (normalized by the square root of its time length) or the cell Largest move on its trajectory. It also contains the average Track curvature, which we designed as follows for each trajectory: for each position $t$, an orthogonal regression is performed on $\{(x_i, y_i)| i \in \{t, \ldots, t + \Delta_t\}\}$ using orthogonal distance regression ($\Delta_t = 10$). The mean curvature of the trajectory is the average of all regression sums of squares.

3.2.3 Averaged local features

Finally, two features are averaged local features, which are the cell MSD and its Mean signed turning angle.
controls from the same batch

The existence of two feature subgroups within this group can be explained by the correlation between track time lengths and features, which was verified by a principal component analysis (PCA). The first seven principal components explain 95% of the variance.

3.3 Statistical procedure

HT screening data are organized in batches of experiments which includes a set of HT screening data. Each batch includes a set of experiments with regard to single cell motility. False discoveries are controlled using the Benjamini–Hochberg procedure (Benjamini and Yekutieli, 2001). This procedure is repeated n times to ensure that the final p value of an experiment i does not depend on the choice of a specific subset of control experiments in its batch. Here is its formalized description:

1. Compute a sample of statistic (3) under null hypothesis from control–control comparisons.

For each batch bi, for k in \{1, \ldots, C_b (C_b - 1) / 2\}, where \(C_b\) is the number of controls of batch b that passed the quality control

a. Randomly split the control experiments in two groups \(A_{b,k}\) of cardinal 2, and \(B_{b,k}\) of cardinal \(C_b - 2\).

b. For each control j of \(A_{b,k}\), compute the statistic \(S_{bk,j}^0\) (3) by comparing it with the pooled group of controls \(B_{b,k}\).

2. Compute statistics from experiment–control comparisons. For computation time feasibility, only n = 5 repetitions corresponding to n splits of the controls set \((A_{b,k}, B_{b,k})\) are selected on each batch for experiment–control comparisons.

a. For each repetition k in \{1, \ldots, n\}:

i. For each experiment i belonging to a batch bi, compute the statistic \(S_{bi,i}\) by comparing it with the pooled group of controls \(B_{b,k}\).

b. Combine distinct iterations: To be conservative, we chose the following approach:

\[ S_i = \max_{k \in \{1, \ldots, n\}} S_{bk,i} \] (4)

3. For each experiment i, compute the p value \(p_i\):

\[ p_i = \max\left( \frac{\text{card}\{\{b, k, j\} | S_{bk,j}^0 \geq S_{bi,i}\} }{\text{card}\{\{b, k, j\}\}} \frac{1}{\text{card}\{\{b, k, j\}\}} \right) \]

4. For each experiment i, compute the adjusted p value \(p_i^*\) to control the false discovery rate (Benjamini–Hochberg procedure; Benjamini and Yekutieli, 2001)

3.4 Validation on a simulated screen

3.4.1 Screen simulation

To evaluate the performance of our workflow on data for which the ground truth is known, we designed a process to simulate an HT screening experiment.

In a first step, five types of single cell movements were designed, in agreement with qualitative observations from the dataset: random, fast random, curved directed, flip directed and stop-and-go (Fig. 5).

Let \((d_i, \phi_i)\) be the polar coordinates of the difference vector \(m_{i-1} - m_i\) of any two consecutive points. For random movement, \(\phi_i\) is chosen at random and the distance \(d_i = ||m_i - m_{i-1}||\) is drawn from a normal distribution, whose parameters are estimated from the data. The same holds for fast random with increased distance \(d_i\). For the curved-directed movement type, \(d_i\) follows again a normal distribution as for random movement, but the angle is calculated as \(\phi_i + \epsilon\) with \(\phi_i = \phi_{i-1} + \Delta\phi_i\), where \(\Delta\phi\) and \(\epsilon\) follow normal distributions, whose parameters are set manually to visually match some observed trajectories.
Flip directed and stop-and-go are two composite types of movement, where the cells alternate between different states. The dwelling times in the two states are random integers with manually fixed ranges (which can be different for the two states) and are drawn independently for each trajectory. Flip-directed movement corresponds to directed movement ($\theta$ is drawn from a normal distribution) with a 180° flip for every state transition. Finally, stop-and-go movement alternates between slow random movement (where $\phi$ is drawn from a uniform distribution) and fast-directed movement (where $\phi$ is drawn from a normal distribution).

In a second step, we want to simulate movies (controls and experiments), that is, sets of trajectories. For this, we define five movie types with different proportions of single cell movement types (cf supra). Normal movies account both for control movies and experiments in which cell motility is similar to that of controls. They contain on average 80% of random trajectories, and a mix of the four other trajectory types. This reflects our observation that in real data, experiments and controls typically contain all possible types of cell trajectories and that phenotypes are characterized in a shift in percentage. All other movie types contain (on average) 50–65% random trajectories, the rest being completed according to the movie type. For example, movie type fast is composed of 30% of fast random trajectories, 60% of random trajectories and a mix of the three other trajectory types.

The total number of trajectories in each movie was drawn at random from real data in the following way: first, a batch is randomly chosen in the dataset. Then, we assign a permutation of the real trajectory numbers from the experiments of the picked batch to the simulated positions. In this way, we can include potential batch effects in our simulated data. The number of trajectories of each

![Heatmap showing trajectory feature similarities on a subset of the Mitocheck dataset (1.1 million trajectories coming from detected motility hit experiments according to MotIW). The dendrograms were obtained using the Ward method and the Euclidean distance between feature correlations.](https://academic.oup.com/bioinformatics/article-abstract/31/12/i320/215036)

![Simulated trajectories: stop-and-go (green), flip directed (red), random (orange), fast random (purple) and curved directed (blue).](https://academic.oup.com/bioinformatics/article-abstract/31/12/i320/215036)
movement type in each movie is drawn from the corresponding movie type multinomial distribution, where the percentages were defined as described above.

The third step was the simulation of ~50 000 experimental conditions, which were distributed on 130 plates, and performed in triplicate as in Mitocheck experimental setup (Neumann et al., 2010). For the sake of simplicity, triplicates were supposed to belong to the same movie type. On each plate, between 5% and 15% of the experiments were selected to be other than normal movies.

3.4.2 Application to a simulated screen

Our workflow successfully recognized more than 98% of the experiments, as detailed in Table 3.

Our simulation pipeline was also used to estimate how useful the trajectory feature set is to capture the differences between different types of trajectory motion. A total of 500 samples of each trajectory type were simulated, and their features extracted. A PCA was performed, after which we retained the eighth first principal components, which explain ~95% of the dataset variance. Finally, k-means was applied to the dataset with \( k = 5 \).

Many simulation parameters (e.g. each track length) are chosen at random, and k-means’ results depend on its initialization: the procedure was therefore repeated 10 times. The results are presented in Table 3. Although distinguishing trajectory types is subject to some errors, it shows that the whole pipeline is robust enough to identify experiments in which cell motility is significantly different. A reasonable accuracy in terms of trajectory types is also obtained.

3.5 Application to the Mitocheck dataset

After evaluating MotIW on simulated data, we then apply it to the whole genome-wide screen Mitocheck (Neumann et al., 2010), which enables us to identify an ontology of 2D cell trajectories.

In the context of the Mitocheck dataset, the identification of an experiment in which cell motility is significantly different from negative controls leads to the identification of genes which might be involved in its mechanisms.

The application of MotIW to the Mitocheck dataset enabled the identification of the experiments which significantly deviate from controls (5%; 7 153 of 144 909). It amounts to 1 180 genes (out of 11 808) and a migration. As cell population migration during metastasis are thought to be led by some leader cell, it is crucial to study single cell motility. The workflow we have presented in this article allows to quantify single cell trajectories. Therefore its application to large-scale datasets will provide useful insights into the molecular regulation of single cell motility, thereby complementing previous studies on collective cell migration.

The application of this workflow is not limited to RNAi data. In a next step, we are going to apply this workflow to newly generated

| Table 3. Results from the application of MotIW to simulated data |
|------------------|------------------|
|                  | Recall (%) | Precision (%) |
| Outlier experiment detection | 99.2       | 98.9          |
| Outlier condition detection    | 99.5       | 100.0         |
| Trajectory clustering         | 91.4 ± 2.1 | 89.4 ± 4.8   |

This approach did not prove to be successful when applied to pooled trajectories from all experiments, for a wide range of clustering techniques (k-means, Gaussian mixtures models, spectral clustering, fuzzy c-means, kernel k-means—data not shown). It succeeded when all trajectories from the detected experiments were pooled together. Indeed, this small subset contains only experiments which have been selected for being significantly different of controls in terms of single cell motility: it is enriched in rare trajectories. After retaining the first seven principal components (explaining 95% of the variance), k-means was applied to the resulting dataset of ~1.1 million trajectories.

Figure 6 shows the evolution of intra-cluster cohesion and silhouette score with respect to the number of clusters. It points to \( k = 8 \) as being both the best and a good quality clustering on this dataset. Indeed, a break and a maximum are respectively expected in the cluster cohesion and the silhouette score curves at the correct cluster number, if it exists. The cluster characteristics are detailed in Figure 7. Each column in the heatmap corresponds to one cell trajectory, for which the rows show the standard scores of a subset of features.

In the first place, it shows that single cell information can be retrieved by our statistical procedure, which works at the experiment level. Indeed, a result about single cell motility patterns is obtained from experiments which were selected on the basis of their trajectory feature distributions.

In the second place, it shows that there is more than speed for differentiating trajectory types. For example, clusters 2 and 3 present very similar MSDs and Effective space length. However, trajectory curvatures are different: the features Mean curvature and Straightness index are quite distinct between the two clusters. This can be observed in the Supplementary movie, where cells whose trajectory belongs to cluster 2 (green) are much straighter than those belonging to cluster 3 (red). In this video, cells whose trajectory passed the trajectory quality control have a dot, whose color corresponds to its cluster as indicated in Figure 7.

4 Discussion

This article presents a generic methodological framework for studying single cell motility in a HT setup. It combines single cell tracking, newly designed trajectory features and an original statistical procedure. Furthermore, its output could be used to obtain an ontology of cell motility in an unsupervised manner: cell motion types were inferred from the data without using any prior knowledge. We found that clustering procedures might not scale in the presence of great biological variability, as is typically observed in HT datasets. We suspect that this is due to highly unbalanced classes and large biological variability. Taken together, those effects produce continuous looking datasets. However, applying hit detection with the described statistical procedure prior to clustering solved the problem for our trajectory analysis. It may be a promising procedure to apply to other clustering problems in HCS, such as clustering of nuclear or cellular morphologies.

As cell population migration during metastasis are thought to be led by some leader cell, it is crucial to study single cell motility. The workflow we have presented in this article allows to quantify single cell trajectories. Therefore its application to large-scale datasets will provide useful insights into the molecular regulation of single cell motility, thereby complementing previous studies on collective cell migration.

The application of this workflow is not limited to RNAi data. In a next step, we are going to apply this workflow to newly generated
Environmental Toxicology data in order to identify environmentally relevant chemicals which perturb cell motility.

Acknowledgement
Thanks to Michele Sebag for many fruitful discussions.

Funding
This work was supported by the French Ministry of Sustainable Development (A.S.); by the European Community (Systems Microscopy, FP7/2007-2013, 258068 to T.W.) and the European Research Council (SMAC-ERC-280032).

Conflict of interest: none declared.

References
Tan,P.N. et al. (2005) Introduction to Data Mining. 1st edn. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA.