JCI insight

LipidFinder: A computational workflow for discovery of lipids identifies eicosanoid-phosphoinositides in platelets

Anne O'Connor, … , Stuart M. Allen, Valerie B. O'Donnell

JCI Insight. 2017;2(7):e91634. https://doi.org/10.1172/jci.insight.91634.

Resource and Technical Advance **Technical Advance** **Inflammation** **Metabolism**

Accurate and high-quality curation of lipidomic datasets generated from plasma, cells, or tissues is becoming essential for cell biology investigations and biomarker discovery for personalized medicine. However, a major challenge lies in removing artifacts otherwise mistakenly interpreted as real lipids from large mass spectrometry files (>60 K features), while retaining genuine ions in the dataset. This requires powerful informatics tools; however, available workflows have not been tailored specifically for lipidomics, particularly discovery research. We designed LipidFinder, an open-source Python workflow. An algorithm is included that optimizes analysis based on users' own data, and outputs are screened against online databases and categorized into LIPID MAPS classes. LipidFinder outperformed three widely used metabolomics packages using data from human platelets. We show a family of three 12-hydroxyeicosatetraenoic acid phosphoinositides (16:0/, 18:1/, 18:0/12-HETE-PI) generated by thrombin-activated platelets, indicating crosstalk between eicosanoid and phosphoinositide pathways in human cells. The software is available on GitHub (https://github.com/cjbrasher/LipidFinder), with full userguides.

JCI insight

TECHNICAL ADVANCE

LipidFinder: A computational workflow for discovery of lipids identifies eicosanoid-phosphoinositides in platelets

Anne O'Connor,[1] Christopher J. Brasher,[1] David A. Slatter,[1] Sven W. Meckelmann,[1] Jade I. Hawksworth,[1] Stuart M. Allen,[2] and Valerie B. O'Donnell[1]

[1]Systems Immunity Research Institute and Institute of Infection and Immunity, School of Medicine, [2] School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom.

Accurate and high-quality curation of lipidomic datasets generated from plasma, cells, or tissues is becoming essential for cell biology investigations and biomarker discovery for personalized medicine. However, a major challenge lies in removing artifacts otherwise mistakenly interpreted as real lipids from large mass spectrometry files (>60 K features), while retaining genuine ions in the dataset. This requires powerful informatics tools; however, available workflows have not been tailored specifically for lipidomics, particularly discovery research. We designed LipidFinder, an open-source Python workflow. An algorithm is included that optimizes analysis based on users' own data, and outputs are screened against online databases and categorized into LIPID MAPS classes. LipidFinder outperformed three widely used metabolomics packages using data from human platelets. We show a family of three 12-hydroxyeicosatetraenoic acid phosphoinositides (16:0/, 18:1/, 18:0/12-HETE-PI) generated by thrombin-activated platelets, indicating crosstalk between eicosanoid and phosphoinositide pathways in human cells. The software is available on GitHub (https://github.com/cjbrasher/LipidFinder), with full userguides.

Introduction

Lipids are a large heterogenous class of hydrophobic molecules important in health, development, and metabolic disorders, including cardiovascular disease, arthritis, and diabetes (1–4). They account for 30% of most organs but 60% of the brain (w/w). Their analysis, termed lipidomics, increasingly utilizes high-resolution mass spectrometry (MS) and thus requires bespoke informatics tools to process the large volumes of data generated. Lipids comprise up to a third of metabolomic database entries; however, informatics workflows tailored specifically to their analysis, and in particular to discovery of new lipids using high-resolution MS, are not well represented or are only commercially available and cannot be user modified. Currently available lipid-focused workflows — for example Greazy, LipidBlast, LipidView (Sciex), and LipidSearch (Thermo Fisher Scientific) — have been generated primarily for analysis of known lipids, with Greazy, LipidView, and LipidBlast using MS/MS data (5, 6). While LipidSearch can be applied to high-resolution MS analysis as well as MS/MS, it is not designed to mine for novel lipids. Both LipidView and LipidSearch are only available commercially and cannot be modified or improved by addition of code by subsequent users.

Lipids exhibit enormous structural and functional diversity and include many isobaric species, greatly increasing the complexity of their analysis (7, 8). We are only beginning to explore how they change on a global scale in humans, including both known and unknown lipids in individuals over time, and how they are influenced by environmental conditions including diet, health, and time of day, as well as genetic/racial background. Fundamental questions such as the total number and diversity of lipids in mammalian cells remain largely unanswered.

We recently defined the global lipidome in human platelets and how this changes on thrombin activation (9). Significant variation was seen in a small group of healthy donor platelets, with a mean of approximately 5,500 individual species per isolate. Importantly, up to 50% of the detected ions were absent from online databases, indicating significant potential for discovery. In that study, an analytical method that facilitated detection of low-abundance species, often the most biologically important, using long chro-

Authorship note: AOC and CJB contributed equally to this work.

Conflict of interest: The authors have declared that no conflict of interest exists.

Submitted: November 7, 2016
Accepted: February 14, 2017
Published: April 6, 2017

Reference information:
JCI Insight. 2017;2(7):e91634. https://doi.org/10.1172/jci.insight.91634.

matographic analyses combined with high-resolution MS was used. This was needed to minimize ion suppression and matrix effects, and to enable distinction of ions with extremely close molecular masses. An immense volume of data was generated, with subsequent processing being the major rate-limiting step.

A number of freeware and commercial processing tools — for example, XCMS, MZmine, MS-Dial, MetAlign, OpenMS/TOPP, and Progenesis (Nonlinear Dynamics) — are available to process metabolomics high-resolution datasets (10–16). Although sometimes used in lipidomics studies, they were developed for proteomics and metabolomics (17–22). Due to the complexity and size of the lipidomic datasets (typically > 200 MB) these were unable to process them satisfactorily. Many low-abundance but crucial biologically important lipids were not detected, and numerous artifacts remained. Thus, we established an Orbitrap-based workflow, first using SIEVE (Thermo Fisher Scientific) for chromatographic alignment and framing, followed by an in-house–developed Excel tool to better extract peak components, remove adduct ions and contamination, and correct retention times (RTs) (9). This early version was then followed by extensive manual verification. This was extremely labor intensive and not suitable for long-term, high-throughput use. To solve this, we developed LipidFinder in Python, which automates the Excel/manual process, searches three independent online databases to obtain putative identification of lipids, and assigns them to a class based on the LIPID MAPS system. This represents a significant advance, and herein, we compare this with other commonly used approaches, as well as show identification of new lipids in human platelets. The LipidFinder source code is available on GitHub (https://github.com/cjbrasher/LipidFinder), with a full userguide available (Supplemental Userguide). An email address for users to input comments or suggestions is also available (lipidfinder@cardiff.ac.uk).

Results

The following sections give detailed overviews of the approach used by the separate programs that comprise the LipidFinder workflow. We also compare this with three other commonly used metabolomics packages in terms of finding a reference list of lipids and supporting data cleanup. Full details on how to use LipidFinder are provided in Supplemental Userguides; supplemental material available online with this article; https://doi.org/10.1172/jci.insight.91634DS1.

LipidFinder workflow overview. LipidFinder is an open source, platform-independent, Python workflow, developed initially for use with SIEVE to process MS data from lipidomic experiments. SIEVE can align chromatograms, correcting for RT shifts, and can pick out signals, assigning m/z and time brackets called frames. Its output, in comma-separated values format (CSV format), is used as input to LipidFinder. Five separate programs make up the LipidFinder workflow, as shown in Figure 1: (i) Optimiser, an optional parameter optimization step using heuristics that analyzes the users' own data to find optimum values for processing; (ii) PeakFilter, the main peak-finding algorithm, also incorporating contaminant/adduct/isotope removal and RT correction; (iii) Amalgamator, an optional program used to combine data sets from positive and negative runs, where applicable; (iv) WebSearch, used to interrogate online databases to putatively identify lipid species; and (v) FileProcessing, used to merge database results and further clean up the resulting data.

As described in the Methods, data from the open source and platform-independent XCMS may also be integrated into the LipidFinder workflow as an alternative to SIEVE.

Since XCMS currently only uses centroid data, and also because profile mode data is substantially larger (leading to substantial processing and storage issues), we have herein used centroid for analysis with LipidFinder.

Optimiser overview. Optimiser allows users to choose the most appropriate parameter values for their dataset. The success of MS data analysis relies on the correct choice of settings for the various algorithms used. These are often only determined through repeat analysis, adding considerable time and effort, or are chosen manually by ad hoc rules of thumb, lessening the quality of results. With PeakFilter, the peak finding process has the largest impact on data quality. In this, four user-defined parameters govern how the heuristics are applied; peak width, frame proximity, m/z tolerance, and intensity fold difference between adjacent frames. Depending on the lipidomics dataset, optimum settings for these can vary widely. Owing to its rapid processing time, PeakFilter lends itself to automatic parameter selection, and to exploit this, we implemented a "hill climbing" algorithm. This iterative optimization technique involves selection of an arbitrary initial parameter set that is then scored for fitness (Figure 2). Optimized parameter sets are sought by incrementally changing individual parameters and rescoring until further changes do not pro-

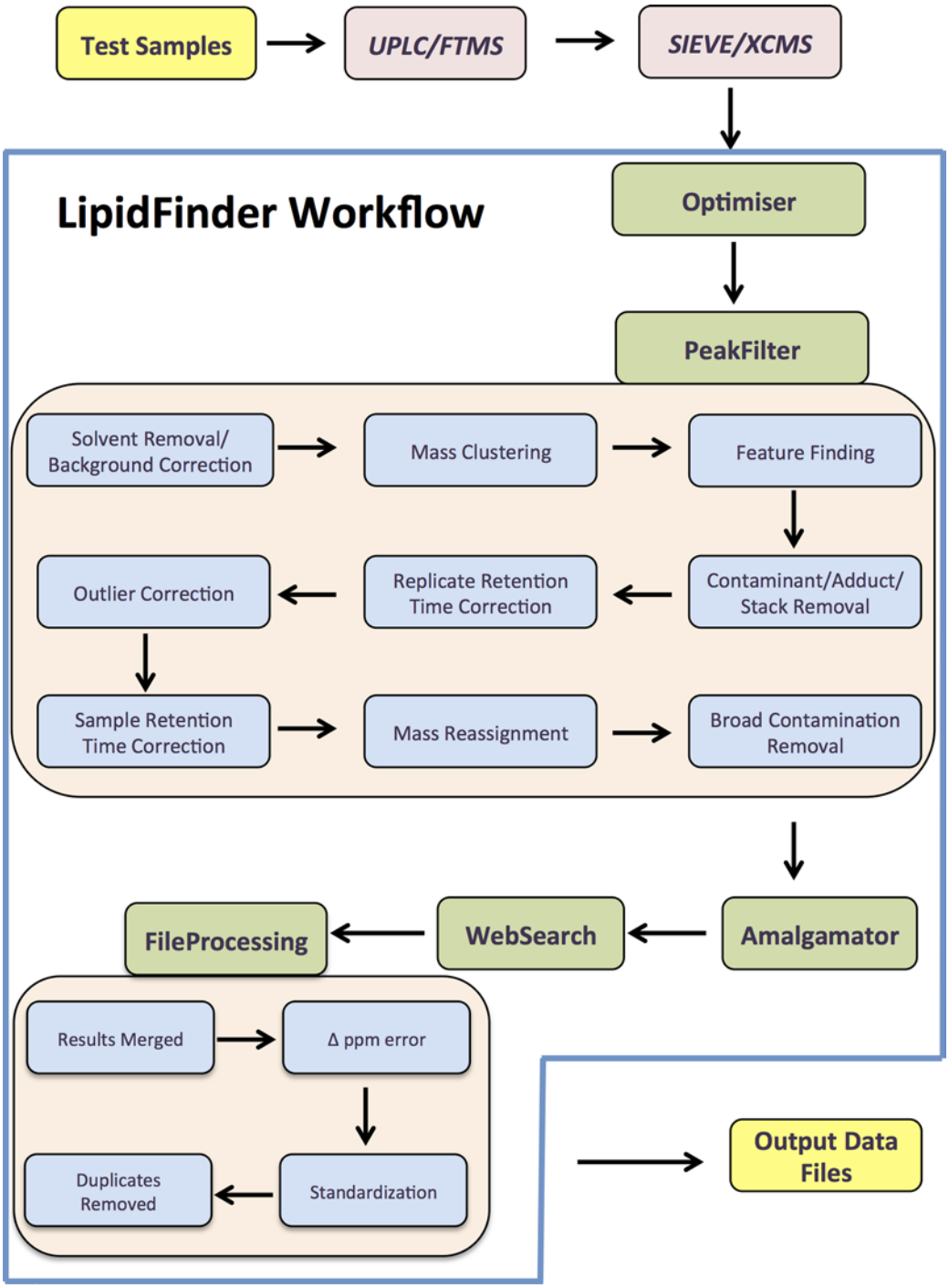

Figure 1. Schematic of the LipidFinder data processing workflow (using SIEVE). Data is first analyzed using UPLC/FTMS, and SIEVE is then fed into the LipidFinder workflow, which incorporates Optimiser, PeakFilter, Amalgamator, WebSearch, and FileProcessing. Data files that retain *m/z*, peak area, retention time, and putative identifications are outputted at the end.

duce improvement. To test the performance of parameters, a representative subset of lipids for a single replicate is curated manually. The suitability of the parameters is then scored against PeakFilter's resulting ability to find them at the correct RT, *m/z*, and intensity. This optional functionality is a significant enhancement over other available data processing tools. Our results that follow below show that incorporating Optimiser into the workflow substantially increases the number of actual/real lipids found. The heuristics are described in more detail in Methods.

PeakFilter overview. PeakFilter can be run with minimal input, with an associated file used to store processing parameters that can be user-modified (Supplemental Data: parameters.csv tab for default values used for this study). These include on/off toggles for a number of optional steps, such as contamination and adduct removal. The mass tolerance value, column type, polarity, and various threshold values can also be recorded.

With long chromatography analyses, small changes in RT between samples often render SIEVE (or other platforms) unable to align peak intensities appropriately. To compensate, PeakFilter corrects for differences between multiple runs, aligning ions with matching *m/z* but slightly different RTs to the time with the highest intensity. This markedly improves alignment and is essential where multiple replicates from different datasets are to be compared for differences in lipidome composition.

A major issue with electrospray ionization (ESI), the commonest mode used for lipids, is the generation of ions arising from either common contaminating ions, in-source fragments of lipids, multiple noncovalent adducts, or solvent background (23–26). To remove these, PeakFilter includes several optional processing steps to clean the data. Unique to our approach, lipid stacks and contamination stacks are

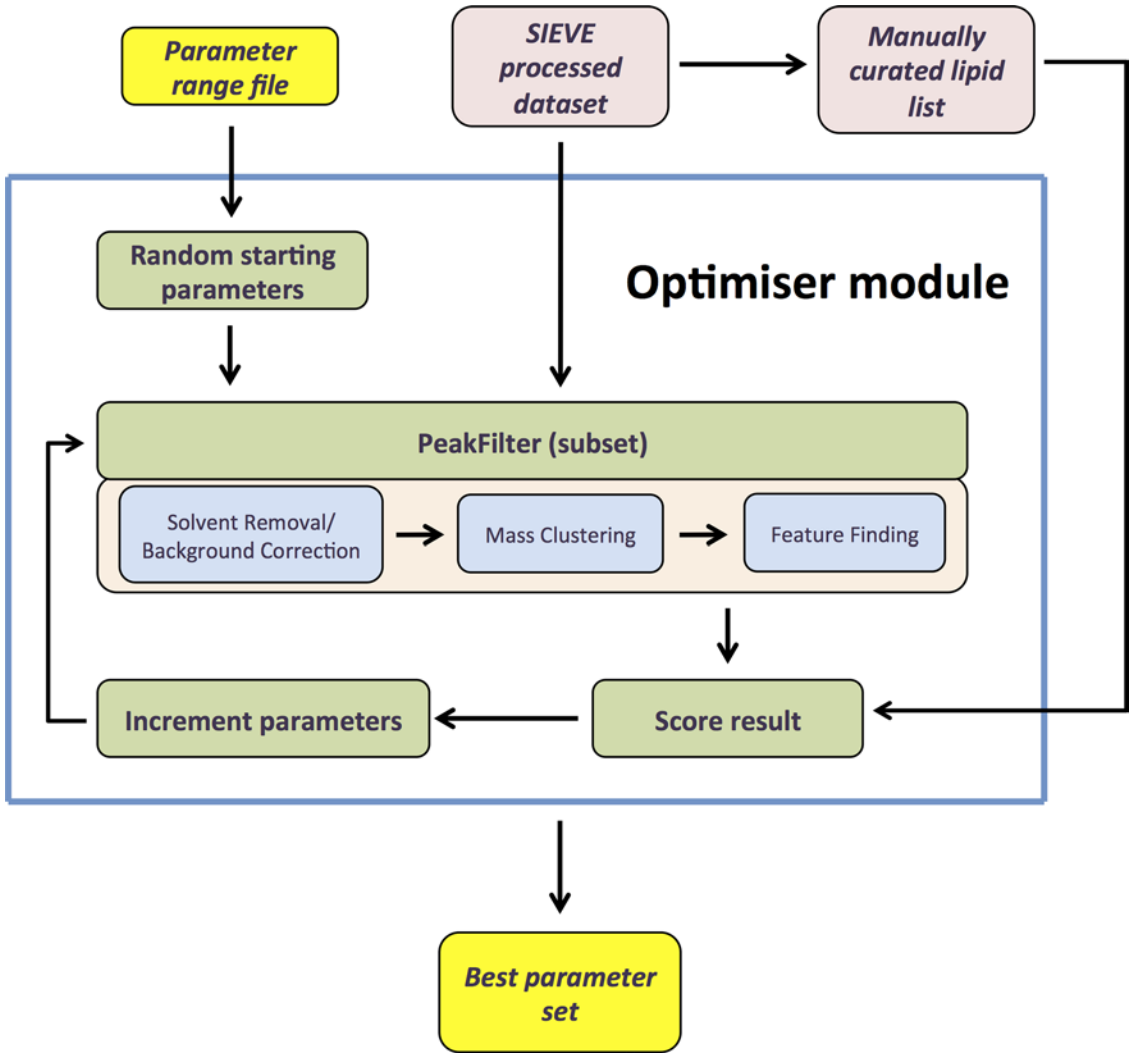

Figure 2. Overview of Optimiser. A range file (max and min list) is inputted, and the module then chooses random starting parameters. These are inputted and tested, then rescored using Optimiser's PeakFilter functionality, until further changes do not produce improvement in peak detection (RT, *m/z*, and intensity). The process uses a representative subset of lipids that has been manually curated from raw data.

removed, where a stack is defined as a series of ions each differing in *m/z* from the next by a fixed mass, a common feature of contaminating ions and adducts in ESI. Lipid stacks elute at the same RT and include noncovalent adducts and in-source fragments. Noncovalent adducts are particularly prevalent in positive ion mode — for example, glycerides, which are commonly detected as both Na^+ or NH_4^+ adducts. In-source fragment ions typical of lipids include loss of water from molecular ions and neutral loss of phospholipid headgroups, especially in positive ion mode. Contaminant stacks are visible as diagonally spaced ions (differing by both RT and *m/z* by defined gaps), often as multiples of the same mass difference. We included published *m/z* values for many known contamination adduct ions but also identified in our dataset by manual interrogation of additional families that were subsequently incorporated into our analysis (Supplemental Data: contaminants.csv and stacks.csv tabs) (23–26). One further new clean-up step is the removal of broad RT contamination, where the same *m/z* values appear across the whole chromatogram, with similar intensities. A detailed description of methods used in these processing steps is provided in Methods and Supplemental Userguides.

Amalgamator overview. If aiming to estimate total number or diversity of lipids in a cell or tissue sample, data from both positive and negative liquid chromatography/ mass spectrometry (LC/MS) analyses needs to be combined. However, many lipids ionize in both modes. To compensate, positive and negative ion mode datasets are processed separately through PeakFilter, then combined using an additional Python

program, Amalgamator. Where positive and negative records match on both *m/z* and RT, the ion with highest intensity is retained. There is also an option to record only the intensity from this ion or the sum of the positive and negative ions together. We use two separate mass differences to compare the positive and negative *m/z* values: 2.014552 amu, for $[M\text{-}H]^-$ vs. $[M+H]^+$ for most lipid classes, and 16.030751 amu for $[M\text{-}CH_3]^-$ vs. $[M+H]^+$ for phosphatidylcholine (PC) lipids. As this may not correct for all lipids, more may need to be added as we refine the methodology. This step can be omitted if not required.

WebSearch and FileProcessing overview. Once lists of *m/z* values have been collated, putative identification using online databases is required. High-resolution MS without fragmentation allows putative (as in proposed) identification only, and when dealing with several thousands of ions, fully validated (with 100% confidence) identification is not possible as a first step. Thus, we typically putatively identify all ions in our samples and then select a smaller group for MS/MS and structural identification based on biological considerations that could include (i) functional relevance, (ii) generation during cell activation/present in disease, and/or (iii) membership of a family of lipids based on *m/z* differences that indicate fatty acid substitutions or oxidation motifs (9). WebSearch automates searches from three primary online repositories — HMDB, LIPID MAPS, and LipidHome — using *m/z* and retains RT and intensity information (27–29). Both LipidHome and LIPID MAPS allow an automated query by WebSearch. Specifically, LipidHome provides a "webservice," while LIPID MAPS allows command line access. In contrast, HMDB is searched using standard web-scraping methods. Similar to LipidFinder, WebSearch uses a parameters file that can be updated as required and allows input of separate *m/z* tolerance values for each database query (Supplemental Data: webSearch_parameters.csv tab). The format of search results is not consistent between databases; thus, FileProcessing combines these into a single results file with lipids automatically categorized into one of the LIPID MAPS classes (30) according to the category map spreadsheet (Supplemental Data: categories_map.csv tab). Additional mappings can be added by the user. Molecules lacking an associated lipid class are categorized as other metabolites. Adduct names are also standardized, and only those listed are retained (Supplemental Data: standardized_adduct_names tab). Duplicate records between databases and records where the mass error is greater than a user-specified tolerance are removed. Online databases comprise a mix of curated and computationally generated lipids, which can give vastly different search results. Advantages and disadvantages of each are covered later under platelet lipid identification.

LipidFinder workflow analysis of platelet lipids. Lipids were isolated from platelets and then analyzed using untargeted LC/Fourier Transform MS (Orbitrap) in negative and positive ion mode (see Methods). In this experiment, two columns were used: one optimized for more lipophilic species (nonpolar, e.g., glycerides, phospholipids, sterol esters) and one for less lipophilic (polar, e.g., fatty acids, eicosanoids). Solvent gradients ran from 30–50 min with full scan (100–900 amu) at 60K resolution. The data was analyzed using the SIEVE/LipidFinder workflow and a CSV output file generated after each step. Optimiser was used to set parameter values. Scatter plots (*m/z* vs. RT) illustrate how PeakFilter cleans and finds real ions for negative ion mode chromatographic data (Figure 3). The entire data set after SIEVE — but before PeakFilter — is shown, followed by sequential cleanup steps, where blue ions remain following each successive cleanup, with black being removed. We note that the number of ions is reduced from 82,853 before PeakFilter to only 3,957 after PeakFilter: a reduction of 95%. This illustrates the utility of our approach to clean up and remove unwanted nonlipid contaminant ions from the dataset.

Comparison of LipidFinder program with XCMS, MZmine, and Progenesis. To determine the effectiveness of LipidFinder's PeakFilter program at cleaning up untargeted datasets, we compared it with three of the most commonly used tools: XCMS, MZmine, and Progenesis. We manually set parameters to achieve the best possible outcomes, although for SIEVE/LipidFinder, we used our hill climbing algorithm (Optimiser). Scatter diagrams of final outputs from each show differences, in particular regarding removal of contaminant ions (Figure 3B). Horizontal rows of ions appear across several plots (same *m/z* value appearing across the time spectrum) but are especially evident in MZmine. Similar "lines" are visible in the XCMS plots. In our LipidFinder workflow, these are mostly eliminated by the broad contamination removal step. Users should manually inspect their data by plotting *m/z* vs. time and visually screening for vertical (lipid stacks), horizontal (RT contaminants), or diagonal (contamination stacks) sets of ions in their dataset. Some diagonal sets will be true lipids — e.g., triglycerides that differ in defined fatty acid chain length and saturation elute in this manner — so careful inspection is needed before routine removal is implemented.

Although each program uses different algorithms, all aim to limit the number of false-positive peaks, while retaining true lipids, a major and complex challenge with large high-resolution lipidomic datasets.

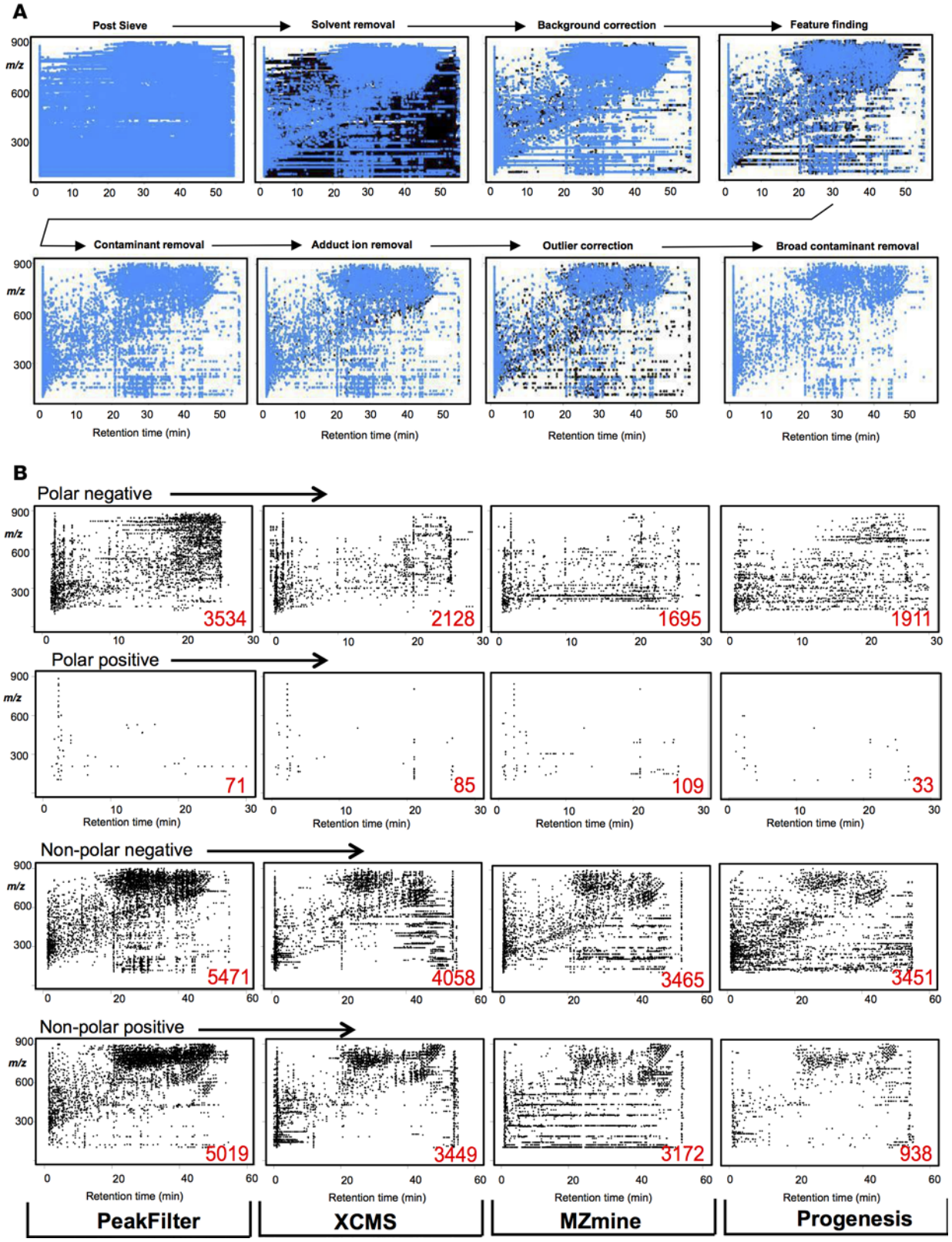

Figure 3. Detection of a list of putatively identified positively and negatively charged lipids by LipidFinder and three commonly used processing packages. (**A**) Table of reference lipids identified by lipid species and category and whether they were detected (green)/undetected (red) by each of the four programs. A series of platelet lipids was manually verified to be present in the Orbitrap dataset, using Xcalibur, and then interrogated for detection using each of the programs, as shown. (**B**) Bar chart summary of **A** results. Data shows the % of lipids in positive/negative ion mode detected by each program, calculated from **A**.

Although the same raw data was used, very different numbers of ions were detected overall using each program. Apart from the polar positive column, where low numbers of ions were found, PeakFilter retained more than each of the other programs (Figure 4). Changing parameter settings would affect these overall numbers, but decreasing the number of ions risks removing potentially important lipids. Crucially, this reflects PeakFilter's ability to mine deep into the data, an important consideration for discovery lipidomics. Currently, LipidFinder is not able to estimate the FDR. In metabolomics, FDR typically uses a target and decoy database where the decoy is made up of scrambled MS/MS spectra (31–33). As we use MS data for LipidFinder, this approach would not be feasible. In late 2017, a second release incorporating fully validated identifications, with MS/MS for up to 500 lipids in plasma, is planned; at that point, we will include FDR in our workflow (34).

To determine the ability of PeakFilter to retain real lipids, relative to other available tools, a reference list of 409 putatively identified lipids, manually verified as present in the raw data, was compared using outputs from each program (Figure 3, A and B). This included both low-abundance lipids (e.g., eicosanoids, prostanoids, and lysophospholipids) present at very low amounts in our platelet extracts and higher-abundance structural species (e.g., phospholipids). PeakFilter (with Optimiser) detected 88% of the reference lipids in positive mode and 84% in negative mode. This was higher than the other programs, with XCMS coming closest (Figure 3B). Overall, this equates to an average detection rate of 86% for PeakFilter, compared with 75%, 57%, and 53% for XCMS, Progenesis, and MZmine, respectively. We note that 4% of the lipids remained undetected by any program, whereas 31% were found by all four. PeakFilter found 44 (11%) of the lipids that were not detected by any other program, while 95% of lipids were found using the combination of PeakFilter and XCMS. Importantly, if Optimiser was not used and parameters were set manually, the PeakFilter detection rate dropped from 86%–68%. This represents a significant advantage of using Optimiser to improve data quality that is unique to our LipidFinder workflow.

Putative identification of platelet lipids using WebSearch/FileProcessing. Before putative identification, positive and negative datasets from PeakFilter were combined using Amalgamator. WebSearch queried these against HMDB, LIPID MAPS, and LipidHome. Results were merged and filtered into lipid groups, and duplicate names were removed using FileProcessing. Using the polar or nonpolar columns for a list of 3,603 or 10,018 *m/z* values, approximately 45%–57% were putatively identified, respectively. For many ions, multiple lipids were identified for a single *m/z* value, in some cases up to several hundred. Thus, the most commonly observed lipid category is used by the program as the category designation. Using this, the majority of polar lipids in human platelets were categorized as fatty acyls or glycerophospholipids, whereas most nonpolar lipids were identified as glycerophospholipids (Figure 5A).

Scatter diagrams, color-coded according to lipid class, are shown to illustrate the range of lipids found in platelets using this approach (Figure 5B). Only 32% polar and 27% nonpolar species were assigned a putative match using all three databases, and there were substantial differences in how each database assigned matches (Figure 5C). This highlights the usefulness of including several databases to gain maximum putative coverage of the lipidome. It also shows that large numbers of ions are not represented. Our recent study on platelet lipids using Excel/manual curation estimated this to be around 50%, somewhat less than that seen here using our Python-based workflow (9). This is likely due to the greater ability of LipidFinder to deep mine and thus uncover more unknown ions.

Online databases comprise a mix of curated and computationally generated lipids. For example, LIPID MAPS includes both, but functionality was recently added that enables the user to search each separately. Similarly, LipidHome contains in silico "theoretical" lipid structures, thus "lipids" that may not exist in mammalian or biological systems are included. HMDB contains all types of small molecule metabolites, including lipids, but these include both detected (measured and confirmed existence) and expected metabolites (known pathways and human intake, but not yet detected in humans) (26). In addition to human intake, the microbiome is another source of potential non–human-derived lipids that could be expected to be detected in analysis of human samples (35). These caveats need to be considered when interpreting database results. As shown herein, the number of hits returned will be considerably greater if computational data is included, and while this increases the likelihood of a match, more inaccurate assignments will also result. The LIPID MAPS curated data only includes mammalian lipids; thus, it may be preferable for initial searches, especially when working with plasma/serum or other human/murine tissues. However, only $[M+H]^+$ or $[M-H]^-$ can be searched using WebSearch. If wishing to focus on known lipids only, using the LIPID MAPS curated list substantially reduces the hits per lipid species. For example, *m/z* value 885.7872

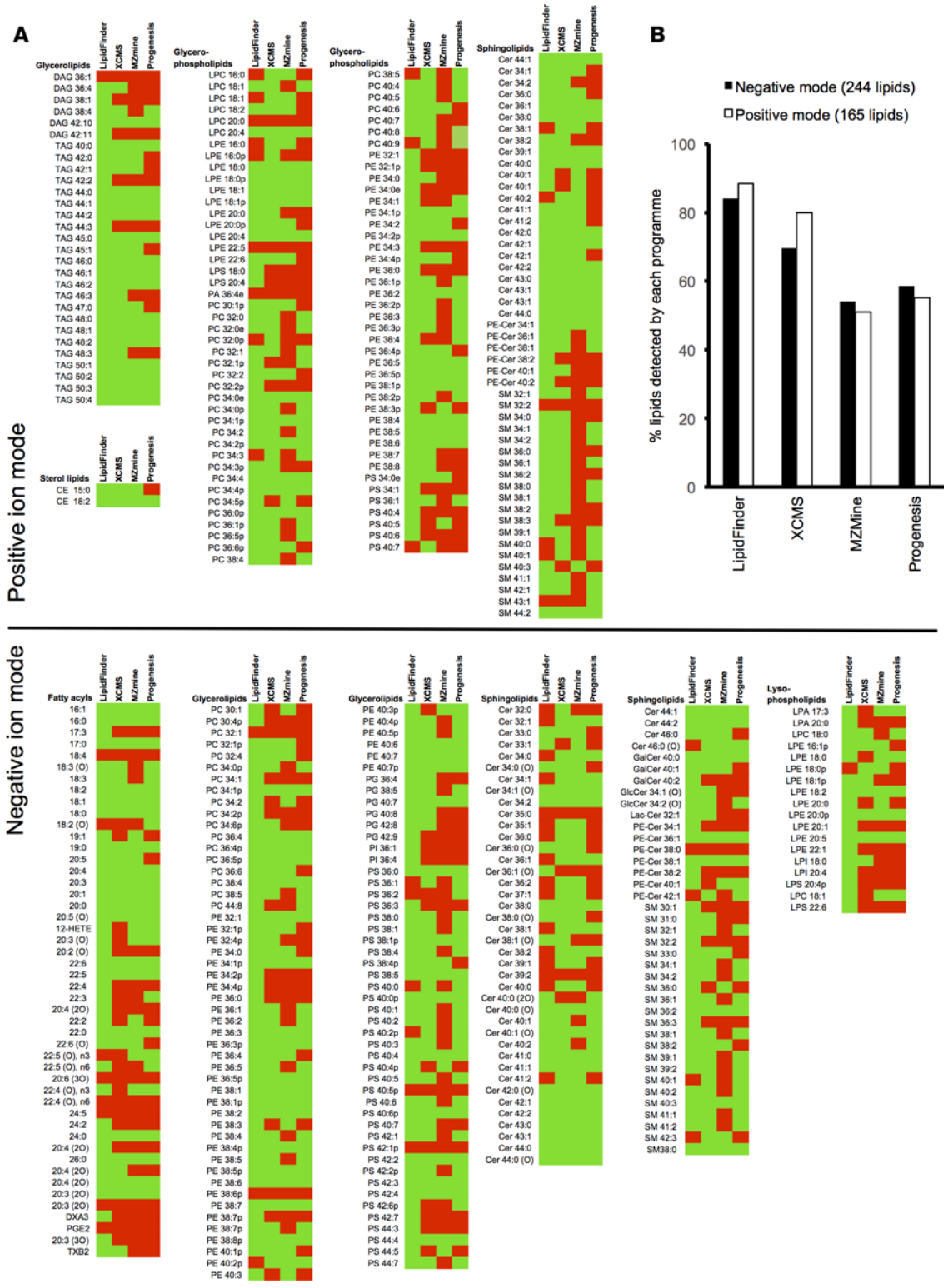

Figure 4. Demonstration of LipidFinder analysis of a dataset of platelet lipids and comparison of performance between LipidFinder and three commonly used metabolomics processing packages. (**A**) Sequential cleanup steps implemented in PeakFilter, after Optimiser parameter setting show removal of large numbers of artifact ions. Scatter diagrams of the PeakFilter output after a selection of steps in the workflow for one sample in nonpolar negative ionization mode. Blue dots indicate *m/z* ions remaining from current step; black dots indicate *m/z* ions from previous step. (**B**) Scatter diagrams of the final outputs from each of the four programs tested, showing the elution of lipids from polar or nonpolar columns, in either negative or positive ionization mode. Red values in bottom right of plots indicate the total lipids plotted. Note the numerous horizontal groups of artifact ions visible in nonpolar negative plots for XCMS, MZmine, and Progenesis.

returned 64 hits when the LIPID MAPS search included curated and computational results, but it returned 2 hits when limited to curated. Additionally, numerous *m/z* values return hits that comprise solely computational data, such as *m/z* 897.7889, which has 65 computational matches. For our nonpolar negative data set of 5,471 ions, 878 had curated LIPID MAPS matches (each with numerous potential IDs), but this almost doubled to 1,624 when computational data was included. Although HMDB stores whether a compound is detected or expected, it is not possible to restrict searches to "detected" data, and this information is not provided in the results file. For example, a search of *m/z* 885.7872 returns 94 hits from HMDB. By manually delving further into each hit separately, "expected" data can be filtered out; however, this is a mammoth and unfeasible task when dealing with very large numbers of potential lipids. For WebSearch, we included three search options, CUR (LIPID MAPS curated only), COM (computational data from LIPID MAPS, HMDB, and LipidHome), and ALL (curated and computational from all three databases).

Structural identification of new families of platelet lipids as eicosanoid-esterifed phosphoinositides. We apply deep-mining, high-resolution approaches to discover lipids likely to be relevant in inflammation and vascular function. Thus, we inspected MS data from lipids extracted from thrombin-activated platelets analyzed as described above and noted a group of ions with *m/z* and RT suggestive of a fatty acid–containing family, e.g., differing by fatty acid chain length and/or saturation (*m/z* 901.5448, 899.5291, and 873.5135). Initial searches suggested these to be phosphoinositides (PI), containing an additional oxygen (38:4, 38:5, and 36:4) (Figure 6, A and B). They were further analyzed using LC/MS/MS and identified as 12-hydroxyeicosatetraenoic acid–PIs (HETE-PIs), with diagnostic daughter ions: *m/z* 179.2 (12-HETE), 319.2 (HETE), and *m/z* 315.2, 241.0, and 153.1 for the PI headgroup, and 255, 281 and 283 for *sn1* fatty acids 16:0, 18:1, and 18:0 for parent *m/z* 873, 899, and 901, respectively (Figure 6C). They were not detected in resting platelets and elevated robustly on thrombin activation (Figure 6, D and E). The absence of internal daughter ions for other HETE positional isomers and the presence of single peak for each on LC/MS/MS confirms that they are enzymatically generated via the platelet 12-LOX isoform from endogenous substrate on pathophysiological activation of platelets (Figure 6). PI is a low-abundance but critically important signaling lipid, and molecular convergence between PIs and eicosanoid pathways has not been observed before. We previously reported that platelets acutely esterify 12-HETE to the highly abundant phosphoethanolamine (PE) and PC phospholipids, generating procoagulant species through changing the interaction of the headgroup with membrane-binding proteins (36). Similarly, the presence of HETE in PI may alter the orientation of the inositol headgroup in the cell membrane, leading to changes in how this critically important signaling lipid interaction becomes phosphorylated to PI and subsequent interactions with kinases or PH domain–containing proteins. Indeed, PIs are well-known mediators of platelet function (37–39). Next, PI can become glycosylated to form glycosylphosphatidylinositol (GPI), which — via covalent reactions — forms anchors for proteins in membranes (40). In both cases, the presence of HETE in PIs or GPIs could alter their biology through changing their physical interactions with membrane hydrophobic compartments.

Summary. Fundamental questions remain unanswered regarding the diversity and overall number of lipids in mammalian cells and tissues, and informatics tools to specifically analyze datasets generated from lipidomics of healthy and diseased tissues are lacking. Existing informatics tools were not developed specifically for lipids, nor do they attempt to remove artifacts sufficiently for robust estimates of global cellular lipidomes to be undertaken. To this end, we developed a new Python workflow, LipidFinder, to automate data cleanup and peak finding for lipidomics and to putatively identify the resulting ions. We compared LipidFinder with three widely used metabolomics tools and showed it was superior at deep mining, specifically detecting a reference list of lipids and returning more genuine hits, while removing contaminating/artifact ions. LipidFinder detected both low- and high-abundance ions, reflecting its utility for discovery lipidomics. WebSearch highlighted the differences between available online databases and the importance of searching several for complete coverage of available data, while taking care with computational datasets to ensure data quality.

Despite the improvements, several caveats remain. A single workflow will not provide a perfect solution in terms of robust cleanup of all lipidomic datasets, due to the inherent complexity and diversity of these datasets. Thus, researchers should consider specific needs and how to interpret outputs. Testing data quality using curated lists of predicted lipids is essential. Furthermore, analysis of low-abundance and often highly biologically important ions requires a high degree of MS sensitivity, as well as robust and reproducible chromatographic separation. Herein, two columns were used; however, molecules

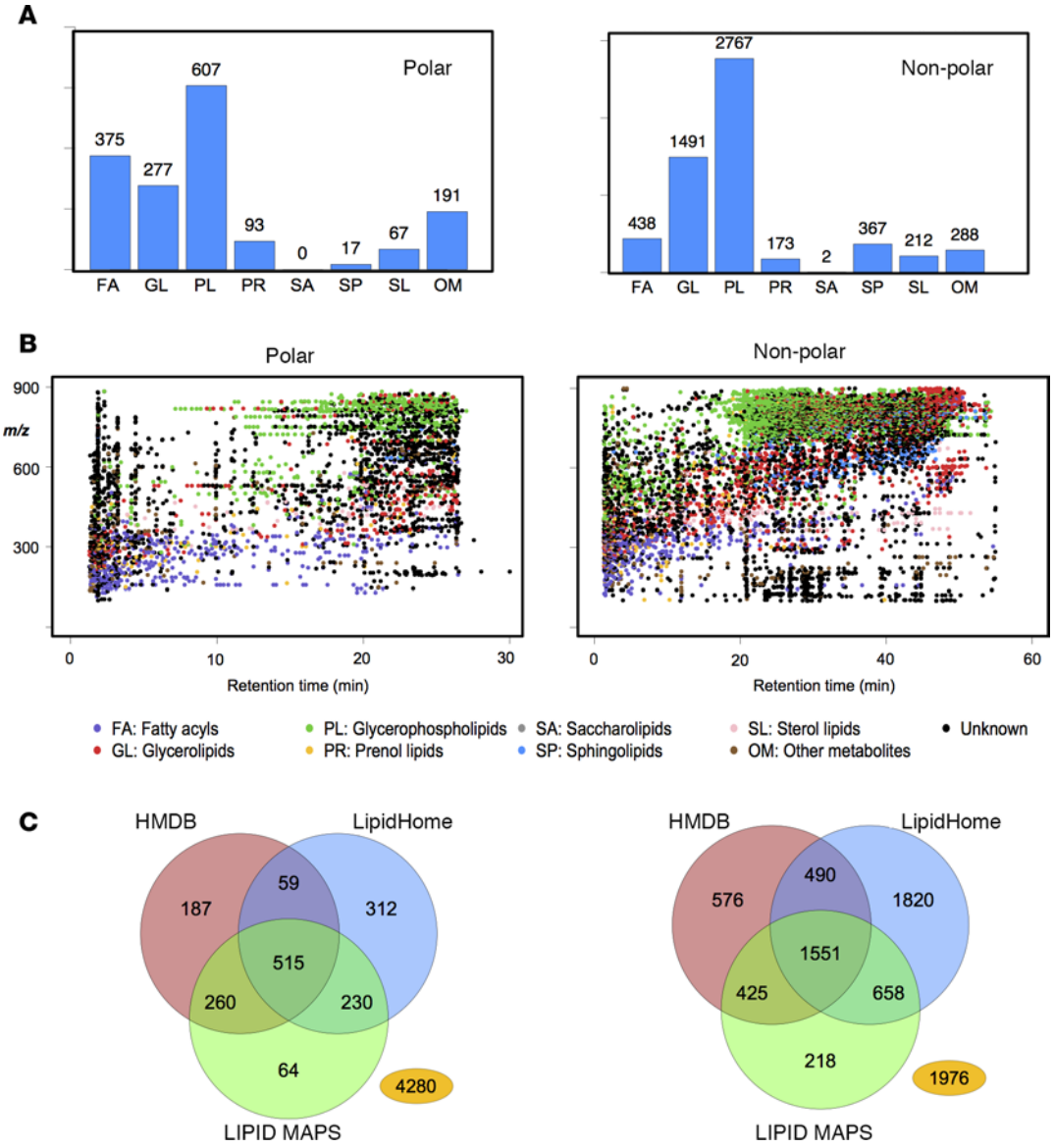

Figure 5. Putative identification of LipidFinder results and comparison of database searches. (**A**) Bar charts showing predominant lipid molecular species in platelets are phospholipids. Each ion was classed using FileProcessing according to the most prevalent hits from three databases into LIPID MAPS categories. (**B**) Scatter diagrams of the LipidFinder output showing elution of lipids from polar or nonpolar columns, in either negative or positive ionization mode and color-coded by lipid category. (**C**) Venn diagrams showing the utility of using several databases for putative identifications. Distribution of hits across three different databases found using WebSearch is shown. The number in yellow represents the number of lipids not given any putative match. Computational and curated data from all databases was used.

detected on both then needed to be identified and removed. In our previous study, this was done manually — an extremely time consuming activity (9). Recently, we implemented a single 60-min separation using a C18 Accucore column (fused core material 2.7 μm, 2.1 mm x 150 mm), removing the need for duplicate removal (unpublished data).

Lipidomics increasingly aims to compare large numbers of cellular or disease cohort samples in order to identify differences that describe sample sets. This focuses on discovery of biomarkers either for diagnostic use in personalized medicine or for further mechanistic study. High-throughput analysis either using no or shorter chromatography runs is often preferred in order to enable large sample numbers to be analyzed and thus increase statistical power. While designed primarily for discovering new lipids, the LipidFinder workflow can also be applied in a high-throughput context, increasing cleanup in order to decrease the likelihood of false positives. Future releases will incorporate additional features

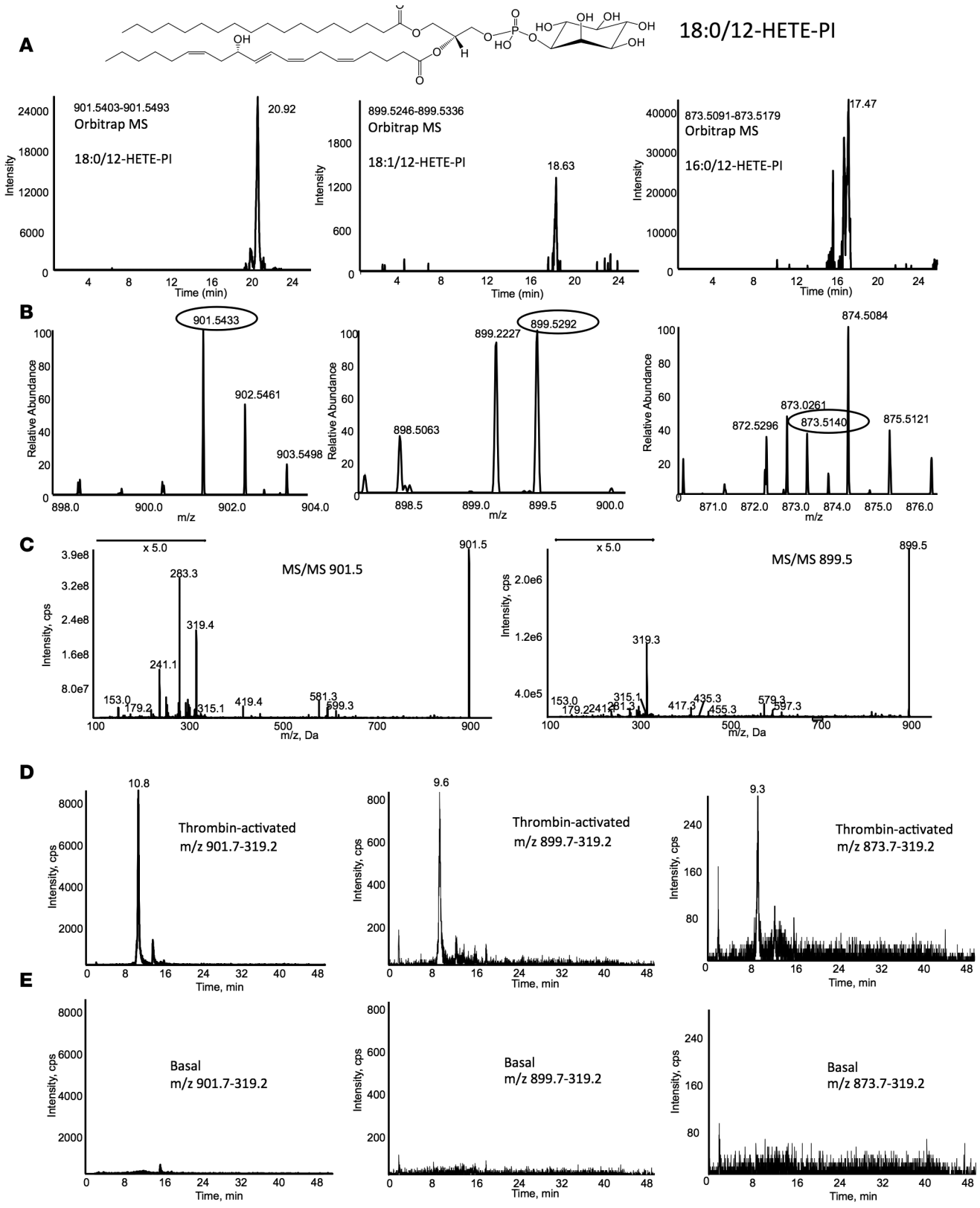

Figure 6. MS identification of eicosanoid-phosphoinositide lipids generated by platelets. (**A**) Orbitrap MS of HETE-PIs in platelet lipid extracts at 60,000 resolution (at *m/z* 400) showing elution of ions with *m/z* values corresponding to 18:0, 18:1, and 16:0/12-HETE-PIs. (**B**) MS of the putative HETE-PI ions. Single ions are shown at expected *m/z* values, circled, for the corresponding lipids in **A**. (**C**) MS/MS of the two most abundant HETE-PIs. Ions from 12-HETE (*m/z* 179, 319), PI headgroup (153,241, 315), and sn1 fatty acids (283 and 281 for 18:0 and 18:1, respectively) are seen. (**D** and **E**) LC/MS/MS separation of thrombin-activated (**D**) or basal (**E**) platelet lipid extracts monitoring MRM transitions corresponding to HETE-PEs. HETE-PIs are only detected following activation of platelets.

to ensure that LipidFinder remains a cutting-edge informatics tool for lipidomics. We point out that, as our code is deposited on GitHub, others can freely download, use, and modify without restriction, including adding new modules.

We show using platelets that many more mammalian lipids (and likely metabolites) remain to be identified, in particular low-abundance species of likely biological importance. Thus, the new workflow fills an identified gap in existing tools required for discovery of new lipids. The major challenge now lies in deciding which lipids to invest effort in structurally characterizing, and this needs to be based on biological considerations, e.g., potential to act as biomarkers of disease or signaling mediators. Structural identification of new lipids is an immense task, since many are present at very low levels, complicating their purification and analysis by biophysical methods such as NMR. Further tools and improvements to existing workflows will be required to facilitate this enormous challenge.

Methods

Human platelet isolation

Blood was collected from three genetically unrelated donors that were free from nonsteroidal antiinflammatory drugs for at least 14 days. Blood was collected into acid-citrate-dextrose (ACD; 85 mM trisodium citrate, 65 mM citric acid, and 100 mM glucose) at a blood/ACD ratio of 8.1:1.9 (v/v) and centrifuged at 250 *g* for 10 minutes at room temperature (22°C). Platelet-rich plasma was collected and centrifuged at 900 *g* for 10 minutes, and the pellet resuspended in Tyrode's buffer (134 mM NaCl, 12 mM $NaHCO_3$, 2.9 mM KCl, 0.34 mM Na_2HPO_4, 1.0 mM $MgCl_2$, 10 mM HEPES, and 5 mM glucose, pH 7.4) containing ACD (9:1, v/v). Platelets were washed by centrifuging at 800 *g* for 10 minutes and then resuspended in Tyrode's buffer at a concentration of 2×10^8 cells/ml. Where activated, they were prewarmed for 1 min at 37°C, with 1 mM $CaCl_2$, before addition of 0.2 U/ml thrombin for 30 min.

Lipid extraction

Lipids were extracted by adding a solvent mixture (1 M acetic acid/propan-2-ol/hexane; 2:20:30, v/v) to platelets at a ratio of 2.5 ml of solvent mixture/ml platelets in 10 ml extraction vial and vortexed for 30 seconds. Hexane (2.5 ml) was added and then vortexed and centrifuged (500 *g* for 5 minutes at 4°C) to recover lipids in the upper hexane layer. Aqueous samples were reextracted by addition of 2.5 ml hexane. The combined hexane layers were dried in a RapidVap (Labconco) at room temperature.

LC/MS of lipids

Two methods for untargeted global analysis of lipids were used: one for lipophilic (nonpolar) species and the other for nonlipophilic (polar) species. All analyses were at 25°C. Polar-LC used a Spherisorb ODS2 column (150 × 2.1 mm, 3 μm particle size) with solvent gradient of mobile phase A (water/acetonitrile, 75:25, v/v, 1 mM ammonium acetate and 0.1% glacial acetic acid) and B (methanol/acetonitrile, 60:40, v/v, 1mM ammonium acetate and 0.1% glacial acetic acid) at 0.4 ml/min over 30 min. The elution gradient of B (%) was: 50%–90% for 20 min, held at 90% for 5.1 min and equilibrated at 50% for 4.9 min. Nonpolar-LC used a Hypersil GOLD C_{18} RP UPLC column (150 × 2.1 mm I.D., 1.9 μm particle size) was used with mobile phase A (acetonitrile/water, 50:50 v/v, 1 mM ammonium acetate, 0.1% glacial acetic acid) and B (iso-propanol/acetonitrile, 70:30 v/v, 1 mM ammonium acetate, 0.1% glacial acetic acid) at 0.4 ml/min over 55 min. The elution gradient of B (%) over time was: 35%–50% for 10 min, 50%–66% for 6 min, 66%–76% for 22 min, and 76%–96% for 10 min, held at 96% for 4.5 min and then equilibrated at 35% for 2.5 min. MS conditions were as follows: HESI-II temperature 350°C, N_2 as drying gas, sheath gas flow 52 arbitrary units, auxiliary gas flow 17 units, capillary temperature 320°C, spray voltage ± 3.5 kV and S-lens RF level 69.8% or 65.60%, respectively, for positive and negative ion mode. MS spectra were acquired using a high-resolution Orbitrap Elite (60,000 at 400 amu) in full-scan Fourier transform mass spectrometry (FTMS) mode over 100–900 *m/z* (for both lipid separation and ion polarity methods) in centroid mode. Mass spectra were acquired in centroid mode. Samples were analyzed using multiple chromatographic runs over a mass range of 100–900 *m/z* to incorporate polar and nonpolar species in both positive and negative ionization modes. Four sets of spectral data (Thermo RAW file format) were generated: nonpolar positive, nonpolar negative, polar positive, and polar negative. Samples were analyzed in random order, using a blank solvent injection every fifth or sixth sample. At the start, a one-off column conditioning was done

with two runs of solvent blank followed by five runs of unrelated platelet extract and then two runs of solvent blank using the same solvent gradient. We used four technical replicates per sample.

LC/MS/MS of PIs

Lipids were separated on a C_{18} Luna, 3 μm, 150 mm × 2mm column (Phenomenex) gradient of 50%–100% B over 10 min followed by 30 min at 100% B (Solvent A: methanol/acetonitrile/water, 1 mM ammonium acetate, 60:20:20. Solvent B: methanol, 1 mM ammonium acetate) with a flow rate of 200 μl/min. Electrospray mass spectra were obtained on a Q-Trap instrument (Applied Biosystems 6500 Q-Trap) operating in the negative mode. Products were analyzed in the MRM mode monitoring transitions from the parent to daughter ion of *m/z* 319.2 (HETE $[M\text{-}H]^-$) every 75 ms with a collision energy of –50 V.

Data processing

Data was input to each of the four programs: SIEVE (version 2.2)/LipidFinder (version 1.0), MZmine (version 2.14.2), XCMS (version 1.44.0), and Progenesis QI (version 2.2). Parameter settings used with each program are given in the program_parameters tab of Supplemental Data. Although each uses different methods, similar values were used where parameters were comparable. Resulting peak lists were obtained from each of the programs.

SIEVE/LipidFinder

The raw data was first processed using SIEVE for peak alignment, isotope removal, and data extraction. The SIEVE parameters were optimized separately for each column, i.e., for lipophilic or nonlipophilic lipid species, and the results were exported as a CSV file. Parameter optimization was run to determine the most optimal parameter settings for use with LipidFinder. This required an input file of manually curated lipids to the Optimiser program. The output from SIEVE was then processed through PeakFilter. Positive and negative ion mode chromatograms were analyzed separately for comparison with the other programs, but also combined using the separate Amalgamator program for use with WebSearch, which was used to putatively identify *m/z* values. This resulted in one large data file per database, with multiple matches per *m/z* value. The results were last processed through FileProcessing, merging the database results, standardizing category and adduct names, and removing duplicate records between databases. Each step is described in detail below.

Optimiser

The PeakFilter process quickly distinguishes and quantifies lipid-like features from contaminants in LC/MS datasets that have been prealigned and processed using SIEVE and is a major component that significantly improves data quality. Underpinning this process is the Peak Finding step. This uses a heuristic approach to determine which frames are to be included in each peak and if the peak's profile qualifies it as a lipid-like peak. The peak-finding algorithm uses just four parameters, and depending on the experiment, the optimal values can vary greatly. Therefore, sensible parameter selection is crucial, as these parameters have a huge impact on the quality and validity of the lipid profile generated downstream. Supplemental Figure 1 illustrates how three of these parameters; namely *peakAdjacentFrameMaxRT*, *peakMinFoldCutOff*, and *peakMaxRTWidth* provide a peak shape template. The other parameter, *mzSizeErrorPPM* governs the maximum intra-peak mass tolerance allowed.

The peak-finding process in PeakFilter is rapid (<3 seconds for single replicate of 60,000 frames), making it suitable for local search optimization where peak finding will need to be performed repeatedly. Here, we implemented a hill climbing algorithm with random restarts. Although relatively simple, hill climbing has several requirements to enable iterative progression to the optimum parameter set for the experiment. Firstly, the peaks produced by a particular parameter set need to be scored. Next, new, previously unscored parameter sets that represent an incremental change in the current best-scoring parameter set need to be generated. Finally, the process should stop when it is not possible to make a previously unscored incremental change to the current best parameter set (top of the hill). Detailed information on Optimiser steps are provided below.

*Scoring mechanism***.** A representative subset of lipids for a single replicate is curated from the SIEVE data into a target set on an individual basis by manual inspection of chromatograms and spectra. The lipid peaks in this target file are described by their intensity, *m/z*, and RT. The suitability of a parameter set is scored

against its ability, when passed into PeakFilter, to find the lipids in the target set at the correct intensity, *m/z*, and RT. The scoring mechanism works by representing the curated peak (CP) geometrically as the center point of a cuboid in 3-dimensional space (Supplemental Figure 2). The CP's *m/z* tolerance range is the *m/z* ± the maximum *m/z* error. The RT tolerance range is CP's RT ± the inter frame distance. The intensity tolerance is the CP's intensity ± its intensity. If a peak is found within the cuboid, then this is regarded as a hit and is scored according to its vector distance from the CP; the closer to the CP the nearer to 100% the score. The distance of each side of the cuboid is normalized to 2; thus, the maximum distance from the center to a corner is Ö3. Subtracting the vector distance of the found feature from the center away from Ö3 gives the score for that particular target. This can be formally represented by the equation: $Individual\ Score = \sqrt{3} - \sqrt{([\%\ proximity\ RT]^2 + [\%\ proximity\ m/z]^2 + [\%\ proximity\ intensity]^2)}$. The total score for a candidate parameter set is found by performing this scoring for each target and then averaging the all the target scores. This gives a score between 0 and 1 for the candidate parameter set. Formally: $(\Sigma_{[i=1]}^{N} \times Score_i)/N$.

Deciding manually if a peak in a lipidomic dataset is a feature of interest is a straightforward process for an experienced researcher. Conversely, manually identifying every lipid peak from chromatograms and spectra in a typical lipidomic dataset would be prohibitively time consuming. Thus, we manually identify a small representative subset of lipid peaks (~100) for a single replicate and use this as a target list whereby different combinations of peak-finding parameters can be compared by scoring their ability to find the targets

Hill Climbing algorithm. The range and granularity of each parameter (known as parameter dimensions) to be tested by the hill climbing algorithm can be specified for each run and are stored in an input file that is imported at run time along with target file, SIEVE-processed file, and PeakFilter general parameters. The algorithm progresses as follows: a list of candidate parameter values (CPV) is generated from the parameter dimensions' list (PDL). An initial, random starting parameter set is selected from the CPV and set as the best current parameters (BPC); this parameter set is scored. The first parameter is now set to the lowest candidate value for this parameter in the CPV, *v1* (provided it was not set to this initially); the other parameters remain at their current values (BPC), and this new parameter set is scored. If it is higher than the BPC, then it is set to be the new BPC and the next parameter, *v2*, is processed in the same manner (provided it was not set to this initially). This is continued until all candidate values have been tried and scored. At this point, the second parameter is assigned its *v1* (again, provided it was not assigned this value initially). The algorithm proceeds as for the first parameter for this and all the subsequent parameters. We define the set of parameter sets produced for a parameter as its 1opt parameter set, and we define a complete pass through each parameter in this manner as a 1opt cycle. The procedure continues repeating 1opt cycles until one is completed with no improvement in score for any parameter. A parameters-visited list is maintained to provide a record of the process and is also used throughout the process to ensure previously seen parameter sets are not rescored (41). When hill climbing completes, the parameter set that provided the best score against the target set should be used as the input for the PeakFilter run. Supplemental Figure 3 illustrates this process in a flow chart.

PeakFilter

Program parameters (LipidFinderData.py). PeakFilter reads parameter values from the *parameters.csv* file (provided in Supplemental Data), generated using Optimiser when data is preprocessed with SIEVE.

Raw data import (LipidFinderData.py). XCMS or SIEVE CSV files are inputted. In the case of this example, we used SIEVE. Where there are multiple files per dataset (e.g., several chromatography runs), PeakFilter combines these into one overall file. This situation can arise if a large dataset is being analyzed, and SIEVE analysis needs to be broken down into discreet RT windows. In the case of SIEVE, the raw data has been aligned and split into frames, which are discreet windows of RT and *m/z* derived for the sample set. Peaks of interest in SIEVE-derived data are made up of both single and multiple frame lipids (lipid peaks that span two or more SIEVE frames). When the raw data has been processed with XCMS, the concept of frames is retained, but each lipid-like peak is made up of a single frame only. Either all positive or all negative ion mode runs from one dataset can be processed at one time.

Quality control (QC) samples (qcCalcs.py). This provides the user with information on the reproducibility of the chromatography and MS in datasets where QC samples are used. In the parameters.csv file (Supplemental Code File), two parameters can be set a lower relative standard deviation (RSD) cut off (*QCLowRSD*) and a higher RSD cut off (*QCHighRSD*). The mean and RSD of QC samples for each frame

are calculated, the number of frames lower than each of these cut offs is counted, and the ratio between them calculated (*QCLowRSD:QCHighRSD)*. This is reported on screen for the user.

Solvent removal (solventsCalcs.py). If there are three or more solvent sample replicates, they undergo outlier correction at the frame level to eliminate intensity outliers (as per sample replicate outlier corrections section). The mean intensity of the remaining solvent replicates (after outlier correction) for each frame is calculated. Frames where every replicate of every sample replicate is not at least a specific fold level above the mean solvent level (*solventFoldCutOff*) are removed in their entirety. Remaining frames have their corresponding mean solvent intensities removed from their replicate intensities. Solvent removal can be toggled on or off with the parameter *removeSolvent*.

Low-intensity removal (solvent.py). Replicate intensities below the threshold value (*intensitySignificanceCutOff*) are set to zero. If all replicates for all samples in a frame are zero, then this frame is removed. Mass clustering, feature clustering, and feature cluster peak analysis steps are only performed on SIEVE data, since peaks have already been integrated at this point in XCMS data.

Mass clustering (clustering.py). Lipid peaks may be seen to elute over several SIEVE frames (e.g., isobaric lipids), with these frames having the same *m/z*. However, there are often small differences in the *m/z* reported. Here, we use hierarchical clustering to group similar *m/z* values (within a specified tolerance) into isobaric groups, called mass clusters. The tolerance is the sum of a variable ppm mass error based on the *m/z* (*mzSizeErrorPPM*), plus a fixed error determined by mass accuracy of the machine (*mzFixedError*). Frames within a mass cluster are now considered to be of the same *m/z* but are not necessarily the same isomer or even compound (note that below, separate compounds will subsequently be identified and labeled as such).

Feature clustering (clustering.py). SIEVE frames within each mass cluster are then sorted by RT. Contiguous frames, separated by a user-defined RT difference (*peakAdjacentFrameMaxRT*), are then grouped and regarded as the same feature cluster. Thus, a mass cluster may have many feature clusters. Supplemental Figure 4A shows the relationship between mass and feature clusters.

Feature cluster peak analysis (peakFinder.py). An algorithm was developed that identifies lipid peaks from the frames within each feature cluster, where each can contain more than one lipid peak. Starting with the most intense frame within a feature cluster, the intensity fold differences (*peakMinFoldCutOff*) and RT distances (*peakMaxRTWidth*) of adjacent frames are compared to build a profile of the peak. This is repeated until all lipid peaks within each feature cluster have been found. Sharper, narrower peaks are classed as lipid-like features; wider flatter peaks are discarded as contamination (Supplemental Figure 1).

Mass contaminant removal (contaminantRemoval.py). Common, well-known electrospray contaminating *m/z* ions are removed in this step. Frames are removed from the dataset that match (within a tolerance) the list of contaminant masses found in the file *contaminants.csv* (Supplemental Data: contaminants.csv tab). RT is not considered; if a mass match is found at all it is removed. Mass contaminant removal can be toggled on or off with the parameter *removeContaminant*.

Adduct ion removal (contaminantRemoval.py). A user-maintained list of noncovalent adducts and their mass differences are listed in the *adducts.csv* file (Supplemental Data: adducts.csv tab). Where an adduct is found and the RT matches that of the parent ion (e.g., $[M+H]^+$, the *m/z* with the highest intensity is retained as the feature of interest and the other *m/z* intensity set to zero). Adduct removal can be toggled on or off with the parameter *removeAdduct*.

Stack removal (*contaminantRemoval.py*). We define a stack as a series of ions, each differing in *m/z* from the next by multiples of a fixed mass. There are two types of stacks: lipid stacks and contaminant stacks. Lipid stacks typically elute at the same RT, while the RT of contaminant stacks increases with increasing *m/z*. The mass differences for both lipid and contamination stacks are stored in the file *stacks.csv* (Supplemental Data: stacks.csv tab). Stacks may contain a number of gaps between multiples, these are governed by *maxStackGap*. Stack removal can be toggled on or off with the parameter *removeStack*.

Replicate RT correction (rtCorrect.py). In our experience, despite the use of SIEVE, frame intensities can still be commonly misaligned. PeakFilter automatically corrects for differences in RT between multiple runs. Specifically, sample replicates with the same *m/z* but slightly different RTs are aligned to the time with the highest intensity. This alignment can only happen if intensities are moving from a sparse to a more populous frame with respect to zero intensity count, and the intensity that is moving must be within a user-defined number of standard deviations (*rtCorrectStDev*) of the destination frame (Supplemental Figure 4B).

Sample replicate outlier correction (*outlierCorrect.py*). The RSD of the technical replicates within a sample an individual frame is calculated. If the variation is too high, then an attempt to reduce below the threshold is made by considering the highest deviation intensity. If this falls outside 2 standard deviations of the other replicates, it is removed and the process repeated with the remaining replicates. If the RSD still exceeds the threshold, then the variation cannot be attributed to any individual value, and as such, all the replicates for the sample are set to zero for that frame. The number of replicates that can be set to zero is fixed and varies on the replicate count per sample: zero if less than 4 replicates, 1 for 4 or 5 replicates, and 2 for 6 or greater. If no more corrections are allowed and the RSD is still too high, then again the whole replicate set for a sample is set to zero (Supplemental Figure 4C).

Sample mean calculation (sampleMeansCalc.py). The non-zero means of each frame's sample replicates are calculated and inserted as a new column for each sample.

Mean RT correction (rtCorrect.py). This works in exactly the same way as Replicate RT correction (from replicate retention time correction section) but corrects misaligned RTs for the sample means (from sample mean calculation section) only. This function should be used where there are no sample technical replicates.

Mass reassignment (reassignMass.py). Following Step F, each mass cluster contains several SIEVE frames with near identical *m/z* values. Next, a single *m/z* is assigned per mass or feature cluster (depending on user preference), using the *m/z* value of the highest intensity SIEVE frame. Note that multiple feature clusters may occur within the same mass cluster, where feature clusters are separated by different RTs(Supplemental Figure 4A). Mass reassignment in feature clusters can be toggled on by using *featureLevelMassAssignment*, the default is mass cluster assignment.

Broad RT contaminant removal (broadContaminant.py). An additional source of contamination occurs where a contaminant elutes at similar intensities continuously across the chromatogram at the same *m/z*. Multiple peaks within the same mass cluster that have similar intensities are automatically removed. However, high statistical outliers are considered to be genuine lipid-like peaks and are retained (individual higher-intensity peaks in a sea of lower-intensity peaks). Starting with the largest statistical outliers (*broadContRSDCutOff* and *broadContrtSDCutOff*), intensities are removed recursively in an attempt to find the set with least variance; a minimum number of intensities must remain (*broadContminPoints*), otherwise no correction can take place (Supplemental Figure 4D). XCMS already includes algorithms for finding and identifying features of interest and reporting them as peaks, and so cannot be used with Optimiser. When used with XCMS data, feature clustering and feature cluster peak analysis steps in Peakfilter should use parameters that coincide with the XCMS parameters such that they do not interfere with the XCMS peak assignment. Depending on the report format, similar approaches could be used for integrating other platforms, adding in or disabling elements of the workflow as appropriate.

MZmine

The standard data processing workflow from the MZmine 2 manual was used (http://mzmine.github.io/documentation.html). First, imported files were filtered using baseline correction and a mass list of detected ions for each scan generated. This was used to build a chromatogram for each mass, and individual peaks were then separated with the deconvolution algorithm. Isotopes were removed and adduct ions identified. RTs were normalized between peak lists, and the peaks in different samples were then aligned. Finally, gap filling was carried out (gaps in peak list rows are filled in by generating a new peak using the largest data point of each scan within the *m/z* and RT range for that row), and the resulting peak list was exported as a CSV file. Parameters used are listed in Supplemental Data: program_parameters.

XCMS

The raw data was filtered and peaks identified using the centWave algorithm, considered the most suitable for high-resolution data in centroid mode (42). Matching peaks across samples were then grouped, so that RT drifts between runs could be corrected. The peaks were regrouped after RT correction, as the original groups were no longer valid, before missing peaks in samples were then filled in. Parameter settings were used that enabled both wide and narrow peaks to be found. Parameters used are listed in Supplemental Data: program_parameters. The resulting data was saved as the peak list output.

Progenesis

Raw data was imported, and automatic processing was selected. We manually selected the most suitable

run as the alignment reference and performed automated peak picking. Parameters used are listed in Supplemental Data: program_parameters. We then checked that the alignment was appropriate both visually and by reviewing the alignment scores. The resulting peak list was exported as a CSV file.

Comparing outputs from all 4 workflows. Using our data, a reference list of 532 ions, corresponding to putative lipids detected in human platelets, was compiled, covering a range of *m/z* values (129.0924–896.6382 Da) in both positive and negative ionization modes, and including both low abundance (eicosanoids) and high abundance (phospholipids) lipids (34). All were manually verified as genuine peaks in raw data chromatograms and RTs recorded. These were then compared against the output peak lists from each program. To be considered a match, *m/z* values had to be within a combined ppm mass error (20 ppm), plus fixed error (0.0005) of the reference value RTs also had to match within a tolerance of 1 minute of the reference RT for inclusion.

Study Approval

For platelet studies, informed consent was obtained, and the study was conducted under Cardiff University School of Medicine Ethics (SMREC 12/13).

Author contributions

AOC, CJB, and VBOD wrote the manuscript. CJB, AOC, and SMA designed and wrote the workflow and algorithms. AOC, CJB, DAS, VBOD, and SWM conducted and analyzed experiments. JIH analyzed experiments. All authors read and edited the manuscript.

Acknowledgments

This work was supported by a European Research Council grant (LipidArrays, to VBOD, SMA), the Wellcome Trust (094143/Z/10/Z), a British Heart Foundation PhD Studentship (CJB), and the Cardiff University Research Opportunities Programme (CUROP, to CJB). VBOD is a Royal Society Wolfson Research Merit Award Holder.

Address correspondence to: Valerie B. O'Donnell, Systems Immunity Research Institute, School of Medicine, Cardiff University, Cardiff CF14 4XN, United Kingdom. Phone: 44.2920.687313; E-mail: o-donnellvb@cardiff.ac.uk.

1. Hinterwirth H, Stegemann C, Mayr M. Lipidomics: quest for molecular lipid biomarkers in cardiovascular disease. *Circ Cardiovasc Genet*. 2014;7(6):941–954.
2. Wenk MR. The emerging field of lipidomics. *Nat Rev Drug Discov*. 2005;4(7):594–610.
3. Oresic M, Hänninen VA, Vidal-Puig A. Lipidomics: a new window to biomedical frontiers. *Trends Biotechnol*. 2008;26(12):647–652.
4. Han X. Neurolipidomics: challenges and developments. *Front Biosci*. 2007;12:2601–2615.
5. Kind T, Liu KH, Lee DY, DeFelice B, Meissen JK, Fiehn O. LipidBlast in silico tandem mass spectrometry database for lipid identification. *Nat Methods*. 2013;10(8):755–758.
6. Kochen MA, et al. Greazy: Open-Source Software for Automated Phospholipid Tandem Mass Spectrometry Identification. *Anal Chem*. 2016;88(11):5733–5741.
7. Wenk MR. Lipidomics: new tools and applications. *Cell*. 2010;143(6):888–895.
8. Shevchenko A, Simons K. Lipidomics: coming to grips with lipid diversity. *Nat Rev Mol Cell Biol*. 2010;11(8):593–598.
9. Slatter DA, et al. Mapping the Human Platelet Lipidome Reveals Cytosolic Phospholipase A2 as a Regulator of Mitochondrial Bioenergetics during Activation. *Cell Metab*. 2016;23(5):930–944.
10. Tsugawa H, et al. MS-DIAL: data-independent MS/MS deconvolution for comprehensive metabolome analysis. *Nat Methods*. 2015;12(6):523–526.
11. Smith CA, Want EJ, O'Maille G, Abagyan R, Siuzdak G. XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem*. 2006;78(3):779–787.
12. Pluskal T, Castillo S, Villar-Briones A, Oresic M. MZmine 2: modular framework for processing, visualizing, and analyzing mass spectrometry-based molecular profile data. *BMC Bioinformatics*. 2010;11:395.
13. Lommen A, Kools HJ. MetAlign 3.0: performance enhancement by efficient use of advances in computer hardware. *Metabolomics*. 2012;8(4):719–726.
14. Sturm M, et al. OpenMS - an open-source software framework for mass spectrometry. *BMC Bioinformatics*. 2008;9:163.
15. Orešič M. Informatics and computational strategies for the study of lipids. *Biochim Biophys Acta*. 2011;1811(11):991–999.
16. Codrea MC, Jiménez CR, Heringa J, Marchiori E. Tools for computational processing of LC-MS datasets: a user's perspective. *Comput Methods Programs Biomed*. 2007;86(3):281–290.
17. Pietiläinen KH, et al. Acquired obesity is associated with changes in the serum lipidomic profile independent of genetic effects--a monozygotic twin study. *PLoS ONE*. 2007;2(2):e218.

18. Hilvo M, et al. Novel theranostic opportunities offered by characterization of altered membrane lipid metabolism in breast cancer progression. *Cancer Res*. 2011;71(9):3236–3245.
19. Layre E, et al. A comparative lipidomics platform for chemotaxonomic analysis of Mycobacterium tuberculosis. *Chem Biol*. 2011;18(12):1537–1549.
20. de Grauw JC, van de Lest CH, van Weeren PR. A targeted lipidomics approach to the study of eicosanoid release in synovial joints. *Arthritis Res Ther*. 2011;13(4):R123.
21. Paglia G, et al. Ion mobility-derived collision cross section as an additional measure for lipid fingerprinting and identification. *Anal Chem*. 2015;87(2):1137–1144.
22. Haoula Z, et al. Lipidomic analysis of plasma samples from women with polycystic ovary syndrome. *Metabolomics*. 2015;11(3):657–666.
23. Keller BO, Sui J, Young AB, Whittal RM. Interferences and contaminants encountered in modern mass spectrometry. *Anal Chim Acta*. 2008;627(1):71–81.
24. Tong H, Bell D, Tabei K, Siegel MM. Automated data massaging, interpretation, and e-mailing modules for high throughput open access mass spectrometry. *J Am Soc Mass Spectrom*. 1999;10(11):1174–1187.
25. ESI+ Common Background Ions. Waters Web site. http://www.waters.com/webassets/cms/support/docs/bckgrnd_ion_mstr_lst_4_13_2010.pdf. Accessed March 21, 2017.
26. Barwick V, Langley J, Mallet T, Stein B, Webb K. Best Practice Guide for Generating Mass Spectra. LGC Limited; 2006.
27. Foster JM, et al. LipidHome: a database of theoretical lipids optimized for high throughput mass spectrometry lipidomics. *PLoS ONE*. 2013;8(5):e61951.
28. Wishart DS, et al. HMDB 3.0--The Human Metabolome Database in 2013. *Nucleic Acids Res*. 2013;41(Database issue):D801–D807.
29. Fahy E, Sud M, Cotter D, Subramaniam S. LIPID MAPS online tools for lipid research. *Nucleic Acids Res*. 2007;35(Web Server issue):W606–W612.
30. Fahy E, et al. Update of the LIPID MAPS comprehensive classification system for lipids. *J Lipid Res*. 2009;50 Suppl:S9–14.
31. Keich U, Kertesz-Farkas A, Noble WS. Improved False Discovery Rate Estimation Procedure for Shotgun Proteomics. *J Proteome Res*. 2015;14(8):3148–3161.
32. Jeong K, Kim S, Bandeira N. False discovery rates in spectral identification. *BMC Bioinformatics*. 2012;13 Suppl 16:S2.
33. Elias JE, Gygi SP. Target-decoy search strategy for mass spectrometry-based proteomics. *Methods Mol Biol*. 2010;604:55–71.
34. Quehenberger O, et al. Lipidomics reveals a remarkable diversity of lipids in human plasma. *J Lipid Res*. 2010;51(11):3299–3305.
35. Caesar R, Nygren H, Orešič M, Bäckhed F. Interaction between dietary lipids and gut microbiota regulates hepatic cholesterol metabolism. *J Lipid Res*. 2016;57(3):474–481.
36. Thomas CP, et al. Phospholipid-esterified eicosanoids are generated in agonist-activated human platelets and enhance tissue factor-dependent thrombin generation. *J Biol Chem*. 2010;285(10):6891–6903.
37. Jackson SP, Yap CL, Anderson KE. Phosphoinositide 3-kinases and the regulation of platelet function. *Biochem Soc Trans*. 2004;32(Pt 2):387–392.
38. Min SH, Abrams CS. Regulation of platelet plug formation by phosphoinositide metabolism. *Blood*. 2013;122(8):1358–1365.
39. Purvis JE, Chatterjee MS, Brass LF, Diamond SL. A molecular signaling model of platelet phosphoinositide and calcium regulation during homeostasis and P2Y1 activation. *Blood*. 2008;112(10):4069–4079.
40. Mayor S, Riezman H. Sorting GPI-anchored proteins. *Nat Rev Mol Cell Biol*. 2004;5(2):110–120.
41. Hurley S, Smith DH, Thiel SU. FASoft: A system for discrete channel frequency assignment. *Radio Sci*. 1997;32:1921–1939.
42. Tautenhahn R, Böttcher C, Neumann S. Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinformatics*. 2008;9:504.