

Spatially distinct and metabolically active membrane domain in mycobacteria

Jennifer M. Hayashi^a, Chu-Yuan Luo^a, Jacob A. Mayfield^b, Tsungda Hsu^a, Takeshi Fukuda^c, Andrew L. Walfield^a, Samantha R. Giffen^a, John D. Leszyk^d, Christina E. Baer^e, Owen T. Bennion^f, Ashoka Madduri^b, Scott A. Shaffer^d, Bree B. Aldridge^{f,g}, Christopher M. Sassetti^e, Steven J. Sandler^a, Taroh Kinoshita^{c,h}, D. Branch Moody^b, and Yasu S. Morita^{a,1}

^aDepartment of Microbiology, University of Massachusetts, Amherst, MA 01003; ^bDivision of Rheumatology, Immunology and Allergy, Brigham and Women's Hospital, Harvard Medical School, Boston, MA 02445; ^cDepartment of Immunoregulation, Research Institute for Microbial Diseases, Osaka University, Osaka, 565-0871, Japan; ^dDepartment of Biochemistry and Molecular Pharmacology and Proteomics and Mass Spectrometry Facility, University of Massachusetts Medical School, Worcester, MA 01605; ^eDepartment of Microbiology and Physiological Systems and Howard Hughes Medical Institute, University of Massachusetts Medical School, Worcester, MA 01605; ^fDepartment of Microbiology & Molecular Biology, Tufts University School of Medicine, Boston, MA 02111; ^gDepartment of Biomedical Engineering, Tufts University, Medford, MA 02155; and ^hLaboratory of Immunoglycobiology, WPI Immunology Frontier Research Center, Osaka University, Osaka, 565-0871, Japan

Edited by Michael S. Glickman, Memorial Sloan-Kettering Cancer Center, New York, NY, and accepted by the Editorial Board March 19, 2016 (received for review January 4, 2016)

Protected from host immune attack and antibiotic penetration by their unique cell envelope, mycobacterial pathogens cause devastating human diseases such as tuberculosis. Seamless coordination of cell growth with cell envelope elongation at the pole maintains this barrier. Unraveling this spatiotemporal regulation is a potential strategy for controlling mycobacterial infections. Our biochemical analysis previously revealed two functionally distinct membrane fractions in *Mycobacterium smegmatis* cell lysates: plasma membrane tightly associated with the cell wall (PM-CW) and a distinct fraction of pure membrane free of cell wall components (PMf). To provide further insight into the functions of these membrane fractions, we took the approach of comparative proteomics and identified more than 300 proteins specifically associated with the PMf, including essential enzymes involved in cell envelope synthesis such as a mannosyltransferase, Ppm1, and a galactosyltransferase, GlfT2. Furthermore, comparative lipidomics revealed the distinct lipid composition of the PMf, with specific association of key cell envelope biosynthetic precursors. Live-imaging fluorescence microscopy visualized the PMf as patches of membrane spatially distinct from the PM-CW and notably enriched in the pole of the growing cells. Taken together, our study provides the basis for assigning the PMf as a spatiotemporally distinct and metabolically active membrane domain involved in cell envelope biogenesis.

mycobacteria | cell envelope | lipid biosynthesis | polar growth | membrane domain

Tuberculosis, caused by the infection of *Mycobacterium tuberculosis* (*Mtb*), is a disease that claims about 1.5 million human lives annually (1). The thick, lipid-laden cell envelope of mycobacteria is composed of a plasma membrane, peptidoglycan-arabinogalactan layer, and mycolate outer membrane that are crucial for pathogenicity (2–4). The cross-sectional structure of the five or more distinct layers that form static mycobacterial cell envelope has been elucidated (2, 5). However, any mechanism for lateral elongation of a multilayered structure has not been defined. Most models posit that key components are synthesized inside the plasma membrane, with subsequent transport to outer layers of elongating cell wall (CW) (6). In mycobacteria, the elongation of the cell envelope is restricted to the polar region of the cell (7–10), suggesting the presence of spatiotemporal control mechanisms to supply cell envelope biosynthetic intermediates to this region (6). Indeed, a recent study demonstrated that key biosynthetic enzymes of the peptidoglycan-arabinogalactan-mycolic acid core structure are specifically enriched in the subpolar region of mycobacterial cells (11).

We previously reported membrane compartmentalization in *Mycobacterium smegmatis* (*Msmeg*) (12). Density gradient fractionation of mycobacterial lysate revealed a distinct fraction containing plasma membrane free of the CW (PMf) in addition to a fraction containing the classical plasma membrane tightly associated with the CW (PM-CW). Both of these membranes are composed of major phospholipids such as phosphatidylethanolamine (PE), phosphatidylinositol (PI), and cardiolipin (12). However, the PMf fraction is enriched in specific enzymes related to the biosynthesis of PE and PI mannosides (PIMs) (12), implying its distinct role in phospholipid metabolism. To synthesize phospholipids, mycobacteria use the cytidine diphosphate-diacylglycerol (CDP-DAG) pathway (13), in which phosphatidic acid (PA) is activated to CDP-DAG, and then converted to phosphatidylserine and PI (14). PE is produced from phosphatidylserine by phosphatidylserine decarboxylase (Psd), and this enzyme activity is enriched in the PMf (12). PIMs are made by sequential additions of mannose onto a PI, and AcPIM2 and AcPIM6, containing two and six mannose residues, respectively, are two major products. Although these mature PIM species are distributed in both the PM-CW and PMf, the enzymatic activities from PI to AcPIM2 and from AcPIM2 to AcPIM6 are enriched in the PMf and the PM-CW,

Significance

Mycobacterium is a family of bacteria that includes a number of dangerous pathogens. Arresting the growth of mycobacteria may be possible through the disruption of control points that regulate cell envelope biosynthesis. We demonstrate that *Mycobacterium smegmatis* possesses a spatially distinct biosynthetic membrane domain enriched in the polar growth region of the cell. This membrane domain may act as an organizing center to spatiotemporally coordinate biosynthetic activities during growth in live cells. Thus, our findings provide an important insight into the potential regulatory mechanisms of lipid metabolism in mycobacteria.

Author contributions: J.M.H., C.-Y.L., J.A.M., T.H., T.F., J.D.L., C.E.B., O.T.B., A.M., S.A.S., B.B.A., C.M.S., S.J.S., T.K., D.B.M., and Y.S.M. designed research; J.M.H., C.-Y.L., J.A.M., T.H., T.F., A.L.W., S.R.G., J.D.L., C.E.B., O.T.B., A.M., S.A.S., B.B.A., and Y.S.M. performed research; J.M.H., C.-Y.L., J.A.M., T.H., T.F., A.L.W., S.R.G., J.D.L., C.E.B., O.T.B., A.M., S.A.S., B.B.A., C.M.S., S.J.S., T.K., D.B.M., and Y.S.M. analyzed data; and J.M.H. and Y.S.M. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission. M.S.G. is a guest editor invited by the Editorial Board.

¹To whom correspondence should be addressed. Email: ymorita@microbio.umass.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1525165113/-DCSupplemental.

respectively (12). Furthermore, PimB', the mannosyltransferase that mediates the second mannose addition (15, 16), is specifically associated with the PMf (17). In addition, polyprenol-phosphate-mannose (PPM) is a lipidic mannose donor critical for the synthesis of mannose-containing glycolipids such as AcPIM6, lipomannan, and lipoarabinomannan. PPM synthase, composed of Ppm1 and Ppm2, is essential for survival (18), and its activity is also enriched in the PMf (12). Hence, biosynthetic reactions critical for cell envelope biosynthesis are associated with the PMf. However, whether these enzymes are specifically bound to the PMf and whether the PMf is a spatially distinct membrane in vivo remained undetermined. In the current study, we combined large-scale analytical methods with live-cell imaging. Our data reveal the broader composition of a spatiotemporally distinct membrane domain, demonstrating colocalization of enzymes with the products of lipid pathways in which they operate. These data support the idea that the PMf is an organizing center for the biosynthesis of specific metabolites in mycobacteria.

Results

The PMf Is a Multifunctional Membrane. To broadly understand the protein composition of the PMf, we conducted a comparative proteomic analysis and identified a total of 240 and 626 proteins enriched in the PMf and the PM-CW, respectively (Dataset S1). We used the DAVID gene functional classification tool (19) to reveal enrichment of the transport and metabolic machineries of inorganic ions, amino acids, and carbohydrates in the PM-CW (SI Appendix, Fig. S1). The PM-CW was also enriched in enzymes involved in protein trafficking, energy metabolism, and signal transduction (SI Appendix, Table S1). Our proteomic analysis extends the key conclusion of our previous study that the PM-CW is the classical plasma membrane tightly bound to the CW. In contrast, the PMf was enriched in proteins involved in metabolism of specific cell envelope components (SI Appendix, Fig. S1). As predicted, we observed PimB' (MSMEG_4253), Psd (MSMEG_0861), and Ppm1 (MSMEG_3859) in the pool of 240 PMf-enriched proteins, validating our analysis. Only eight of 240 PMf proteins were predicted to have transmembrane (TM) domains (SI Appendix, Table S2) (20), suggesting that most PMf-associated proteins are peripheral membrane proteins.

Five of the newly identified PMf-associated proteins were chosen for further analysis based on known or predicted function, abundance, protein size, and the presence of a TM domain (SI Appendix, Materials and Methods). They were GlfT2 (MSMEG_6403, UDP-galactosyl transferase), GtfI (MSMEG_0389, glycosyltransferase), a geranylgeranyl reductase (MSMEG_2308), PyrD (MSMEG_4198, dihydroorotate dehydrogenase), and a putative membrane protein (MSMEG_1944), of which the last two have predicted TM domains. We expressed these proteins with a C-terminal HA tag in wild-type *Msmeg* (see SI Appendix, Fig. S15A for the vector design). Western blotting of density gradient fractions showed that all the selected proteins were enriched in the PMf fractions, colocalizing with an endogenous PMf marker PimB' (Fig. 1A and SI Appendix, Fig. S24).

Further analysis of GlfT2 and PyrD showed that both GlfT2-HA and PyrD-HA coimmunoprecipitated with PimB' from crude lysate (Fig. 1B and SI Appendix, Fig. S2B). The coimmunoprecipitation (IP) of PimB' was dependent on the HA epitope tag (SI Appendix, Fig. S2C). MptA (MSMEG_4241), a PM-CW mannosyltransferase involved in the biosynthesis of lipomannan and lipoarabinomannan (17), was not pulled down under the same conditions, and mild detergents disrupted the co-IP of PimB', consistent with the idea that the PMf membrane mediates these protein interactions. Previously, negative-stain transmission electron microscopy (TEM) revealed vesicle-like structures in the PMf fractions (12), termed PMf vesicles. Immunogold TEM revealed both GlfT2-HA and PimB', detected by anti-HA and anti-PimB' antibodies, respectively, on the same PMf vesicles, reinforcing the co-IP results (Fig. 1C). Of

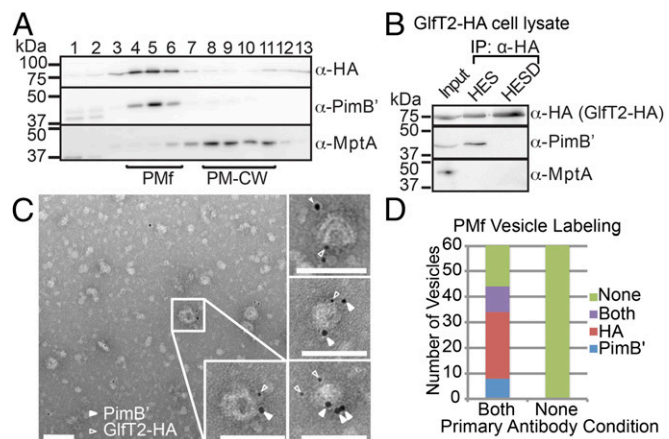


Fig. 1. GlfT2-HA and PimB' are bound to the same PMf membrane. (A) Western blot of density gradient fractions of GlfT2-HA-expressing cell lysate. GlfT2-HA, 74.9 kDa, anti (α)-HA; PimB' (PMf marker), 41.4 kDa, α-PimB'; and MptA (PM-CW marker), 54.3 kDa, α-MptA. (B) Co-IP of GlfT2-HA and PimB' and its disruption by mild detergent. Input is the equivalent amount of lysate used in co-IP experiment. HES (Hepes at pH 7.4, EDTA, NaCl) and HESD (HES with mild detergents) indicate buffers used for IP. (C) Negative-stain immunogold TEM of the PMf from GlfT2-HA expressing cells illustrates colocalization of GlfT2-HA (5 nm gold, open arrowheads) and PimB' (10 nm gold, solid arrowheads). (Scale bar, 100 nm.) (D) Quantification of immunogold-labeled PMf vesicles treated with (as in C) or without (as a negative control) primary antibodies. None, no gold particle detected on vesicle; Both, at least one 5-nm and one 10-nm gold particle detected; HA, one or more 5-nm gold particles detected; and PimB', one or more 10-nm gold particles detected. $n = 60$.

60 randomly chosen vesicles, 44 vesicles were detected by at least one of the antibodies, and 10 vesicles were detected by both (Fig. 1D). Control experiments established the specificity of this detection (Fig. 1D and SI Appendix, Fig. S3 A–E). Taken together, these data indicate the PMf is a multifunctional membrane bound by a specific set of proteins with known functions.

The Refined Proteome Indicates the Roles of the PMf in Lipid Metabolism and Cell Envelope Biogenesis. Many PMf proteins were categorized as having known functions in DNA replication and protein translation, rather than membrane biogenesis (SI Appendix, Fig. S1). In addition, GlnA1 (MSMEG_4290), an abundant glutamine synthetase in mycobacteria (21) that forms a ~600-kDa homododecamer (22), was highly enriched in the PMf (Dataset S1). However, when GlnA1-HA was expressed, it did not colocalize with nor coimmunoprecipitate PimB' (SI Appendix, Fig. S2 A, D, and E). Thus, large cytoplasmic protein complexes, such as GlnA1 and those involved in DNA replication and protein translation, contaminated the PMf fraction, likely as a result of overlapping sedimentation properties.

We established an epitope tag-based purification system to remove contaminants by creating a transgenic *Msmeg* strain with the endogenous *glfT2* gene replaced with an mCherry (mC) fusion gene, *HA-mC-glfT2* (SI Appendix, Fig. S15B). GlfT2 is an essential galactosyltransferase for arabinogalactan biosynthesis (23–25). Transgenic and wild-type strains grew at similar rates (SI Appendix, Fig. S4A), suggesting that HA-mC-GlfT2 replaces the function of the wild-type protein. After confirming the specific PMf localization of HA-mC-GlfT2 (SI Appendix, Fig. S4B), we purified the PMf by density gradient fractionation, followed by anti-HA IP, and performed comparative proteomics on this and an identically treated wild-type sample. Using a stringent criterion of 10-fold enrichment in the PMf from HA-mC-GlfT2-expressing cells, we detected 309 PMf-associated proteins (Dataset S1). Among them, 117 were present in the initial PMf proteome, including previously

characterized PimB', PyrD, Ppm1, Gtf1, Psd, geranylgeranyl reductase, and MSMEG_1944. Importantly, we no longer detected GlnA1 or other suspected cytoplasmic contaminants. Instead, the PMf was reaffirmed as a membrane enriched in proteins involved in cell envelope biogenesis and lipid metabolism, as well as transport and metabolism of secondary metabolites, amino acids, and inorganic ions (*SI Appendix, Fig. S4C*). Notably, we found acyltransferase MSMEG_2934 involved in AcPIM2 biosynthesis, consistent with previous findings that the early steps of PIM biosynthesis take place in the PMf (12). In addition, we found putative DAG kinases (MSMEG_4335 and MSMEG_1920; the orthologs of *Mtb* Rv2252 and Rv3218) (26) and 1-acylglycerol-3-phosphate *O*-acyltransferase (MSMEG_4248; the ortholog of *Mtb* Rv2182c) (27, 28) enriched in the PMf. These enzymes function in two independent pathways of PA biosynthesis, indicating the possibility that the PMf is the site of PA production, a hypothesis investigated in detail later. Collectively, proteomic analysis indicated the PMf as a specialized membrane-protein complex with defined and interrelated metabolic functions.

Comparative Lipidomics Supports Specialized Biosynthetic Function of the PMf. Selective association of lipid biosynthetic enzymes with the PMf suggested that the lipid composition of the PMf and PM-CW might be different. Using an established normal phase HPLC-mass spectrometry (MS) platform and mycobacteria-specific ion databases (29, 30), we completed comparative lipidomics

analysis of the PMf and PM-CW. Biological quadruplicate analyses detected 11,079 molecular events, which represent linked *m/z*, retention time, and intensity values (Fig. 2A). After aligning datasets as paired events with equivalent mass and retention time values and calculating ion intensity ratios for all pairs, we enumerated the molecular events that changed in intensity above a high threshold value (twofold, Benjamini-Hochberg corrected $P < 0.05$). We found 642 and 796 events that were significantly over-represented in the PMf and PM-CW, respectively (Fig. 2A and *Datasets S2* and *S3*). Thus, 13% of all lipids met this stringent change criterion, demonstrating broad differences in lipid composition of the two fractions.

Identification of Altered Lipids. The *m/z* and retention time values embedded in changed events could be used to identify the biochemical identities of key lipids corresponding to the changed events. Because the large number of changed events (1,438) precluded identification of all lipids by collision-induced dissociation MS (CID-MS), we used recently validated methods to prioritize events for biochemical analysis (31, 32). These criteria included detection of multiple isotopes of the same molecular ion (M) or mass intervals characteristic of alkane series with more than one member detected. We further focused on the abundant mycobacterial membrane phospholipids in the MycoMass and MycoMap databases (29), using their reported *m/z* and retention time values for preliminary identifications of matching

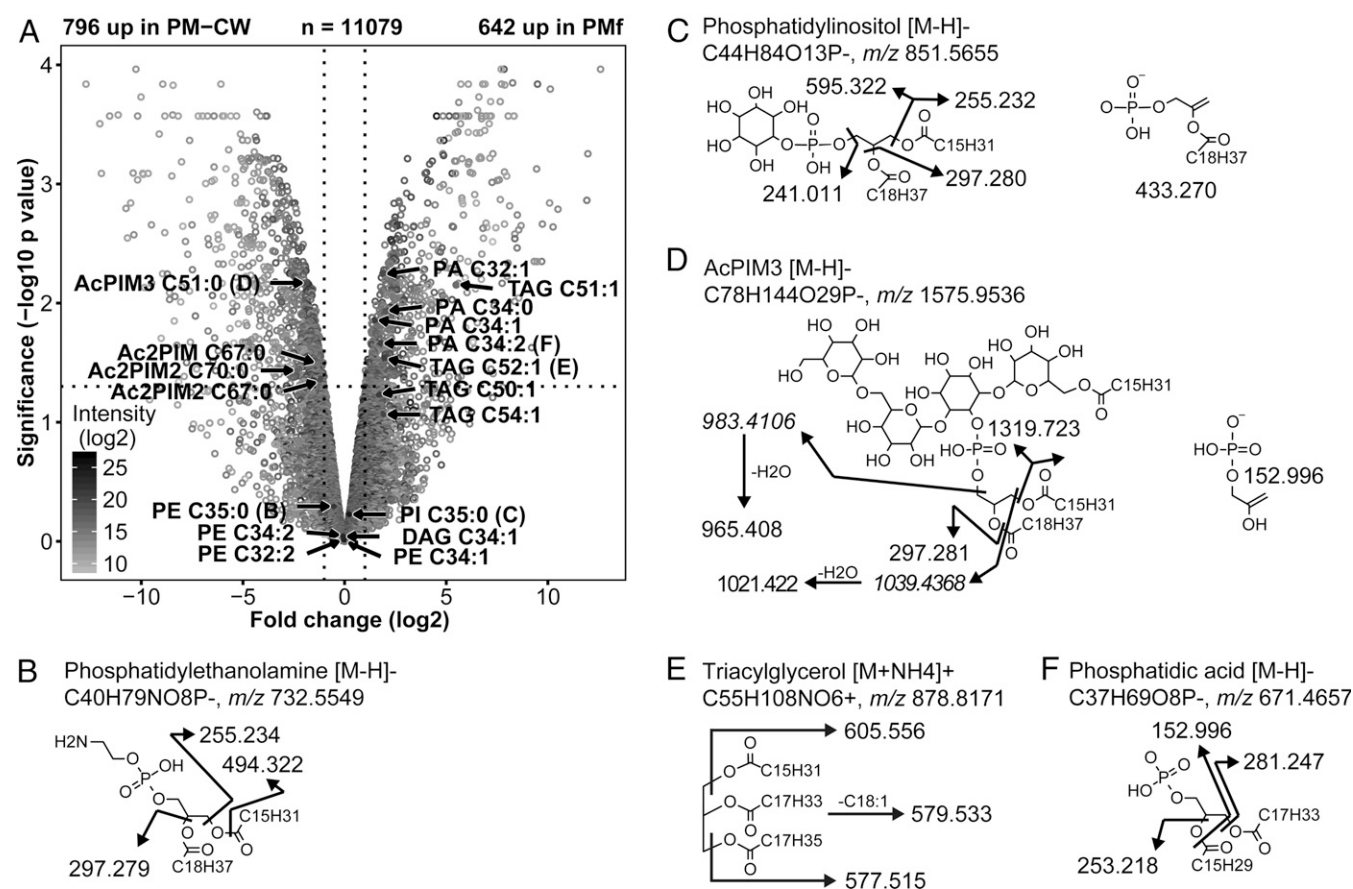


Fig. 2. Lipidomic profiling of the PMf and PM-CW by MS reveals shared and distinct lipids. (A) Volcano plot of ions detected after HPLC-MS. Individual ions (circles) detected in the positive and negative ion modes were combined and plotted based on fold-change and statistical significance in PM-CW versus PMf, using paired samples ($n = 4$). The horizontal dashed line indicates $P < 0.05$ (Benjamini-Hochberg corrected P value); the vertical dashed lines indicate twofold change. The intensity of individual points indicates their relative ion intensity. All lipids indicated by arrows in A were subjected to CID-MS to identify ions of interest, with five representative lipids shown in B–F. PE (35:0) and PI were detected in both fractions (B and C). Monoacyl PI trimannoside (AcPIM3) was PM-CW-enriched (D). TAG (52:1) and PA (34:2) were PMf-enriched (E and F). The chemical structures and fragments detected after tandem MS are shown.

ions (Fig. 2A). Tentative identifications obtained by database matching were validated using CID-MS analysis for at least one member of each class (Fig. 2B–F; *SI Appendix*, Figs. S5–S9 and Table S3, indicated by CID-MS Y; *SI Appendix*, Table S4). Other members of the indicated class were assigned when nearly identical retention times and mass variations typical of acyl chain length or unsaturation were detected (*SI Appendix*, Table S3, indicated by CID-MS N).

Major phospholipid species such as PI (35:0; i.e., 35 carbon chain with no unsaturated bond) and PEs (32:2, 34:1, 34:2, and 35:0) (33) were equally distributed between the PM-CW and the PMf (Fig. 2A and B and *SI Appendix*, Fig. S5 and Tables S3 and S4), consistent with previous TLC analysis (12). Consistent with the previous observations that the biosynthetic steps after AcPIM2 take place in the PM-CW (12), AcPIM3 species were enriched in the PM-CW fraction (Fig. 2A and D and *SI Appendix*, Fig. S7 and Table S3). We also found three acyl forms of Ac₂PIM2 enriched in the PM-CW, suggesting that inositol acylation to produce tetra-acyl PIMs may take place in the PM-CW as well. In contrast, the PMf is enriched in lipid species that are likely involved in core glycerolipid metabolism. Notably, we found four PA species (32:1, 34:0, 34:1, 34:2) enriched in the PMf (Fig. 2A and F and *SI Appendix*, Fig. S9 and Tables S3 and S4). This correlates with the enrichment of PA-producing enzymes in the PMf proteome, as described earlier. Similarly, TAGs (51:1, 52:1) were enriched in the PMf (Fig. 2A and E), and we found a putative TAG synthetase (Tgs) (MSMEG_0290) enriched in the PMf (*Dataset S1*). In contrast, DAG, the substrate for Tgs, was found in both the PMf and the PM-CW, although only one acyl species (34:1) could be reliably detected (*Discussion*). Taken together, despite having a similar major phospholipid composition to the PM-CW, the PMf is uniquely enriched in specific lipid species that correspond to the enzymatic content of the fraction.

The PMf Exists as Discrete Patches in Live *Msmeg* Cells. In addition to the HA-mC-GltT2 strain described earlier, we created a second strain to visualize the PMf by live-imaging fluorescence microscopy. In this strain, endogenous Ppm1 was replaced with the fusion protein Ppm1-mNeonGreen (mNG)-cMyc (*SI Appendix*, Fig. S15B). Ppm1 is the essential catalytic subunit of PPM synthase (18) and was identified in the refined PMf proteome (*Dataset S1*). The transgenic strain grew at a rate comparable to wild type, implying that the fusion protein is functional (*SI Appendix*, Fig. S10A). Furthermore, the fusion protein localized to the PMf during density gradient fractionation (*SI Appendix*, Fig. S10B). We visualized the PMf using these strains expressing either HA-mC-GltT2 or Ppm1-mNG-cMyc, and showed that the PMf formed foci of variable size and intensity throughout the cells, with particularly intense foci at the poles (Fig. 3A). To determine whether the fluorescent patches were artificial aggregates induced by fusion proteins (34), we expressed GltT2 fused with a monomeric mTurquoise2 (mT) in the HA-mC-GltT2-expressing strain (35). We confirmed PMf localization by density gradient fractionation (*SI Appendix*, Fig. S11A) and showed that fluorescent patterns of mT-GltT2-FLAG paralleled HA-mC-GltT2 patterns (*SI Appendix*, Fig. S11B). Previously reported autofluorescence from wild-type *Msmeg* (36) was negligible compared with the fluorescence from mT-GltT2-FLAG expressed in wild-type cells (*SI Appendix*, Fig. S12). These data rule out common artifacts that sometimes occur with protein engineering within bacterial cells and corroborate a method for tracking key PMf markers.

To compare the fluorescent patterns of the PMf and PM-CW, we introduced a PimE-GFP-FLAG expression vector in the HA-mC-GltT2-expressing strain. PimE is a PM-CW-associated enzyme involved in AcPIM6 biosynthesis (37), although it was not identified in the PM-CW proteome, likely because it is a

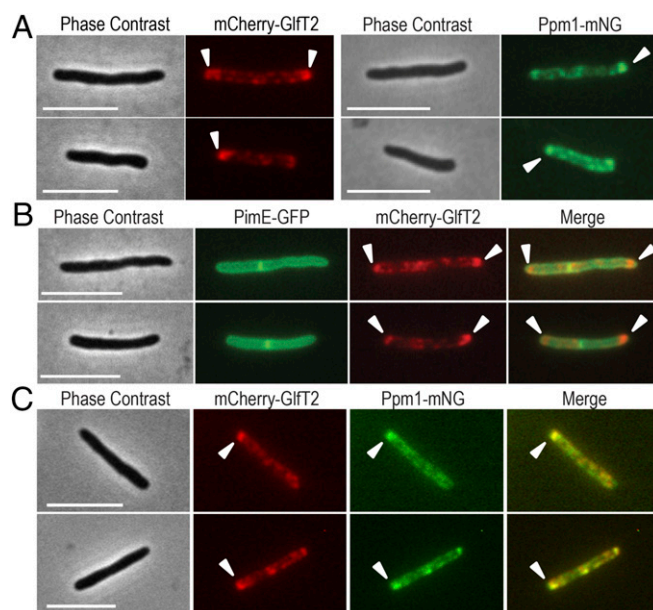


Fig. 3. Live imaging of the PMf markers in *Msmeg*, showing discrete patches of the PMf. (A) HA-mC-GltT2 (Left) and Ppm1-mNG-cMyc (Right) expressing cells showing punctate PMf foci often enriched at the poles of the cells. (B) Coexpression of PimE-GFP-FLAG (PM-CW) and HA-mC-GltT2 (PMf) demonstrates distinct fluorescent patterns. (C) Two PMf-associated proteins (HA-mC-GltT2 and Ppm1-mNG-cMyc) colocalize in single cells. In all panels, arrowheads indicate polar foci. (Scale bar, 5 μ m.) Two representative cells are shown in each panel.

highly hydrophobic membrane protein (*Dataset S1*). We confirmed that PimE-GFP-FLAG was enriched in the PM-CW-containing fraction of density gradients (*SI Appendix*, Fig. S11C). In striking contrast to HA-mC-GltT2 fluorescence, PimE-GFP-FLAG revealed annular fluorescent patterns, in addition to apparent septal fluorescence, consistent with the PM-CW as the classical plasma membrane tightly associated with CW (Fig. 3B).

To examine the colocalization of the two PMf proteins, HA-mC-GltT2 and Ppm1-mNG-cMyc, we generated another transgenic strain in which both endogenous GltT2 and Ppm1 were replaced with HA-mC-GltT2 and Ppm1-mNG-cMyc, respectively. This transgenic strain grew at a similar rate to wild type, implying functional proteins (*SI Appendix*, Fig. S10A). We confirmed PMf localization of both proteins by density gradient fractionation (*SI Appendix*, Fig. S10C and D). We then showed overlapping *in vivo* localization of these two proteins (Fig. 3C), with the Pearson colocalization coefficient of 0.89 ± 0.06 ($n = 20$), which was substantially higher than the coefficient of 0.37 ± 0.16 ($n = 20$) for HA-mC-GltT2 (PMf) and PimE-GFP-FLAG (PM-CW) shown in Fig. 3B. Thus, the PMf is spatially distinct from the plasma membrane in live *Msmeg* cells.

Enrichment of the PMf Is Spatiotemporally Correlated to Actively Growing Poles. Mycobacteria grow from their polar ends, where we found the strongest enrichment of HA-mC-GltT2 and Ppm1-mNG-cMyc foci (Fig. 3). Time-lapse microscopy of the strain expressing both PimE-GFP-FLAG and HA-mC-GltT2 showed that PimE-GFP-FLAG stably associated with the plasma membrane, whereas HA-mC-GltT2 foci continuously associated with the growing poles (Fig. 4A and *Movie S1*). In an alternative approach, we stained the preexisting CW with a fluorescent amine-reactive dye, washed the dye, allowed the cells to grow, and visualized the growth of the unlabeled poles (9). We found that the unlabeled growing poles possessed HA-mC-GltT2 foci

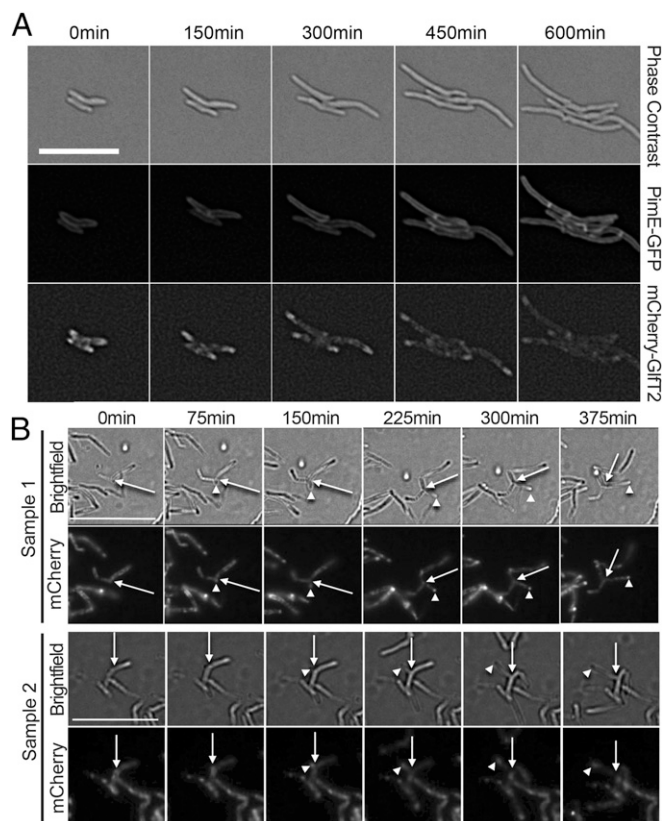


Fig. 4. The spatiotemporal correlation of the PMf and polar CW elongation. (A) Frame shots of time-lapse microscopy (Movie S1) of the PimE-GFP-FLAG/HA-mC-GltT2 dual-expressing cells illustrate the stable annular fluorescence of the PM-CW and the continuous association of the PMf foci at the growing cell poles. Time (minutes) represents the real time since the start of the recording. (Scale bar, 10 μ m.) (B) Two representative time-lapse movies of HA-mC-GltT2-expressing *Msmeg* cells during the recovery phase after transient exposure to DCS (see Movie S2 for sample 1) show the correlation of newly created ectopic polar growth and the PMf foci. The time indicates when the frame shots were taken in relation to the first frame in real time. Arrows, branch point; arrowheads, polar ectopic growth. (Scale bar, 15 μ m.)

(SI Appendix, Fig. S13). These data indicated that the PMf-associated proteins are enriched in the growth pole of the cell.

To further test the temporal correlation of the PMf with polar cell envelope elongation, we induced ectopic polar growth and abnormal branching of the HA-mC-GltT2-expressing cells by transiently treating the cells with D-cycloserine (DCS). We found a fraction of cells initiating growth at ectopic poles during the recovery from DCS-induced growth arrest (Movie S2). Importantly, HA-mC-GltT2 was observed at the site of bifurcation during branching initiation (Fig. 4B) and continuously enriched at the ectopic, DCS-induced growth pole. Thus, our data indicate PMf localization to the poles of mycobacterial cells in two models of growth.

Discussion

Organized lipid microdomains in live mycobacteria were reported in 1999 (38), using fluorescent lipid probes. More recently, cardiolipin enrichment was observed at the septa and poles of actively growing cells with the fluorescent dye 10-N-nonyl acridine orange (39). These studies suggested that the mycobacterial membrane is not entirely homogenous. However, the physiological significance of these observations remained unclear.

In this study, we presented three independent lines of evidence supporting the existence of functionally specialized membrane

domain in mycobacteria. First, proteomic analysis identified more than 300 putative proteins that specifically associate with the PMf. Identification of PimB', Psd, and Ppm1 verified our previous observations (12). In addition, PimB', and Ppm1 are essential enzymes (16, 18), indicating the critical role of the PMf as a site of essential lipid biosynthetic reactions. We confirmed the PMf association of six proteins by density gradient fractionation and performed further analyses on GltT2, Ppm1, and PyrD. GltT2 does not contain predicted TM domains, but rather, its crystal structure implies peripheral membrane association (40). Extending previous studies showing that GltT2 cofractionates with both membrane and CW (24), our data demonstrated stable and specific binding of GltT2 to the PMf membrane. In contrast, the membrane association of Ppm1 is mediated by the formation of a heterodimer with the membrane protein Ppm2 (41), which may dictate PMf localization. We did not detect Ppm2 in our PMf proteome, likely because of the highly hydrophobic nature of Ppm2. The PMf association of GltT2 and Ppm1 predicts that the galactan precursor and PPM, the products of these enzymes, may be found in the PMf lipidome. However, they are transient biosynthetic intermediates and could not be identified by our current analysis. PyrD is a family 2 quinone-dependent dihydroorotate dehydrogenase (42) involved in pyrimidine biosynthesis. It is predicted to be essential in *Mtb* (43) and is hypothesized to interact with the PMf membrane through a predicted TM domain. Whereas precise molecular mechanisms of how these different proteins associate with the PMf membrane remain to be determined, our data indicate that the PMf harbors multiple essential biosynthetic enzymes.

Second, we established that the PMf has a distinct lipid composition from that of the classical plasma membrane. We confirmed our previous observation that PE and PI, major phospholipid species, are equally distributed in both membrane fractions, suggesting that the PMf and the classical plasma membrane are formed by certain ubiquitous phospholipids. In contrast, all four detectable acyl forms of PA are enriched in the PMf, reinforcing the interpretation of the PMf as a metabolically active membrane with specialized function (SI Appendix, Fig. S14). In addition to the PA-producing enzymes, the PMf proteome was enriched in a putative glycerol phosphate acyltransferase (MSMEG_4703) (Dataset S1) that mediates the committed step of PA biosynthesis, further suggesting that glycerolipid biosynthesis initiates in the PMf (SI Appendix, Fig. S14). Interestingly, a DAG species was present in both membrane fractions, suggesting the subsequent reaction of TAG synthesis can take place in either membrane. Among eight *Msmeg* homologs of *Mtb* Tgs enzymes (44), we found MSMEG_0290 enriched in the PMf, consistent with the enrichment of TAG species in the PMf. However, two other Tgs homologs, MSMEG_5242 and MSMEG_1882, were found in the PM-CW, suggesting that TAGs can be made in both membranes but does not accumulate in the PM-CW. Thus, our data may offer a spatially dynamic perspective on the TAG metabolism in mycobacteria (SI Appendix, Fig. S14). Regarding the PIM biosynthesis, we could not identify intermediates such as PIM1 and PIM2 in the lipidome, likely because of their low abundance. Nevertheless, we identified AcPIM2 biosynthetic proteins such as acyltransferase MSMEG_2934 and PimB' in the PMf proteome (SI Appendix, Fig. S14), confirming the previous studies suggesting that AcPIM2 is produced in the PMf (12). The absence of the first mannosyltransferase PimA in the PMf proteome was expected because it is an amphipathic enzyme transiently associating with membrane by weak ionic interactions (12, 45). Because AcPIM2 is found in both membranes, AcPIM2 produced in the PMf must be distributed to the PM-CW. Although further studies are needed to characterize dynamic features of individual metabolic pathways, our current study clearly demonstrates the distinct lipid composition of the PMf with correlative association of specific enzymes.

Finally, we showed that the PMf forms spatially distinct patches with particular enrichment at the poles of live cells. We further showed temporal correlation of PMf markers with the growth of cell poles. Taken together, the current study provided evidence for the PMf as a spatiotemporally distinct membrane domain in which specific biosynthetic reactions take place. Although enriched at the pole, two observations suggest the PMf is not simply a purified cell pole. First, fluorescent patterns of the PMf membrane domain indicate it distributes throughout the cell as discrete patches in addition to polar enrichment. We speculate that these less intense nonpolar patches are involved in cell envelope maintenance, rather than elongation. Second, some pole-associated proteins are not enriched in the PMf. Most notably, DivIVA, a well-known pole-associated protein (11, 46), was enriched in the PM-CW, rather than the PMf (Dataset S1). Taken together, we conclude that the PMf is a membrane domain spatially distinct not only from classical plasma membrane but also from some known polar structures. We suggest that the PMf acts as an organizing center

for multiple biosynthetic enzymes that are critical for cell envelope biogenesis in mycobacteria.

Materials and Methods

Plasmids were made using standard molecular biology techniques (SI Appendix, Fig. S15). Growth of mycobacterial strains, preparation of lysates, and fractionation of lysates by sucrose density gradient were as described (12), with modifications in SI Appendix. Full descriptions of other methods are given in SI Appendix.

ACKNOWLEDGMENTS. We thank Manju Sharma, Matthew Asermely, Kathryn Rahlwes, Julia Puffal, and Stephanie Ha for support, and Dr. Alex Ribbe and Louis Raboin (W. M. Keck Electron Microscopy Center, University of Massachusetts) for assistance with TEM. This work was supported by Human Frontier Science Program Career Development Award, Mizutani Foundation for Glycoscience, and Potts Memorial Foundation (to Y.S.M.), the UMass Graduate School Dissertation Research Grant (to J.M.H.), an Alfred P. Sloan Foundation Research Fellowship and NIH Director's New Innovator Award 1DP2LM011952-01 (to B.B.A.), and NIH Grants U19AI111224 and R01AI049313 (to D.B.M.).

- Zumla A, George A, Sharma V, Herbert N, Baroness Masham of Ilton (2013) WHO's 2013 global report on tuberculosis: Successes, threats, and opportunities. *Lancet* 382(9907):1765–1767.
- Kaur D, Guerin ME, Skovierová H, Brennan PJ, Jackson M (2009) Chapter 2: Biogenesis of the cell wall and other glycoconjugates of *Mycobacterium tuberculosis*. *Adv Appl Microbiol* 69:23–78.
- Mishra AK, Driessen NN, Appelmeik BJ, Besra GS (2011) Lipoarabinomannan and related glycoconjugates: Structure, biogenesis and role in *Mycobacterium tuberculosis* physiology and host-pathogen interaction. *FEMS Microbiol Rev* 35(6):1126–1157.
- Morita YS, et al. (2011) Inositol lipid metabolism in mycobacteria: Biosynthesis and regulatory mechanisms. *Biochim Biophys Acta* 1810(6):630–641.
- Alderwick LJ, Birch HL, Mishra AK, Eggeling L, Besra GS (2007) Structure, function and biosynthesis of the *Mycobacterium tuberculosis* cell wall: Arabinogalactan and lipoarabinomannan assembly with a view to discovering new drug targets. *Biochem Soc Trans* 35(Pt 5):1325–1328.
- Kieser KJ, Rubin EJ (2014) How sisters grow apart: Mycobacterial growth and division. *Nat Rev Microbiol* 12(8):550–562.
- Chauhan A, et al. (2006) Interference of *Mycobacterium tuberculosis* cell division by Rv2719c, a cell wall hydrolase. *Mol Microbiol* 62(1):132–147.
- Thanky NR, Young DB, Robertson BD (2007) Unusual features of the cell cycle in mycobacteria: Polar-restricted growth and the snapping-model of cell division. *Tuberculosis (Edinb)* 87(3):231–236.
- Aldridge BB, et al. (2012) Asymmetry and aging of mycobacterial cells lead to variable growth and antibiotic susceptibility. *Science* 335(6064):100–104.
- Siegrist MS, et al. (2013) D-Amino acid chemical reporters reveal peptidoglycan dynamics of an intracellular pathogen. *ACS Chem Biol* 8(3):500–505.
- Meniche X, et al. (2014) Subpolar addition of new cell wall is directed by DivIVA in mycobacteria. *Proc Natl Acad Sci USA* 111(31):E3243–E3251.
- Morita YS, et al. (2005) Compartmentalization of lipid biosynthesis in mycobacteria. *J Biol Chem* 280(22):21645–21652.
- Nigou J, Besra GS (2002) Cytidine diphosphate-diacylglycerol synthesis in *Mycobacterium smegmatis*. *Biochem J* 367(Pt 1):157–162.
- Jackson M, Crick DC, Brennan PJ (2000) Phosphatidylinositol is an essential phospholipid of mycobacteria. *J Biol Chem* 275(39):30092–30099.
- Lea-Smith DJ, et al. (2008) Analysis of a new mannosyltransferase required for the synthesis of phosphatidylinositol mannosides and lipoarabinomannan reveals two lipomannan pools in corynebacterineae. *J Biol Chem* 283(11):6773–6782.
- Guerin ME, et al. (2009) New insights into the early steps of phosphatidylinositol mannoside biosynthesis in mycobacteria: PimB' is an essential enzyme of *Mycobacterium smegmatis*. *J Biol Chem* 284(38):25687–25696.
- Sena CB, et al. (2010) Controlled expression of branch-forming mannosyltransferase is critical for mycobacterial lipoarabinomannan biosynthesis. *J Biol Chem* 285(18):13326–13336.
- Rana AK, et al. (2012) Ppm1-encoded polyprenyl monophosphomannose synthase activity is essential for lipoglycan synthesis and survival in mycobacteria. *PLoS One* 7(10):e48211.
- Huang W, Sherman BT, Lempicki RA (2009) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nat Protoc* 4(1):44–57.
- Hirokawa T, Boon-Chieng S, Mitaku S (1998) SOSUI: Classification and secondary structure prediction system for membrane proteins. *Bioinformatics* 14(4):378–379.
- Tullius MV, Harth G, Horvitz MA (2001) High extracellular levels of *Mycobacterium tuberculosis* glutamine synthetase and superoxide dismutase in actively growing cultures are due to high expression and extracellular stability rather than to a protein-specific export mechanism. *Infect Immun* 69(10):6348–6363.
- Harth G, Clemens DL, Horvitz MA (1994) Glutamine synthetase of *Mycobacterium tuberculosis*: Extracellular release and characterization of its enzymatic activity. *Proc Natl Acad Sci USA* 91(20):9342–9346.
- Mikusová K, et al. (2000) Biosynthesis of the galactan component of the mycobacterial cell wall. *J Biol Chem* 275(43):33890–33897.
- Kremer L, et al. (2001) Galactan biosynthesis in *Mycobacterium tuberculosis*. Identification of a bifunctional UDP-galactofuranosyltransferase. *J Biol Chem* 276(28):26430–26440.
- Pan F, Jackson M, Ma Y, McNeil M (2001) Cell wall core galactofuran synthesis is essential for growth of mycobacteria. *J Bacteriol* 183(13):3991–3998.
- Owens RM, et al. (2006) *M. tuberculosis* Rv2252 encodes a diacylglycerol kinase involved in the biosynthesis of phosphatidylinositol mannosides (PIMs). *Mol Microbiol* 60(5):1152–1163.
- Mawuenyega KG, et al. (2005) *Mycobacterium tuberculosis* functional network analysis by global subcellular protein profiling. *Mol Biol Cell* 16(1):396–404.
- Larrouy-Maumus G, et al. (2013) Discovery of a glycerol 3-phosphate phosphatase reveals glycerophospholipid polar head recycling in *Mycobacterium tuberculosis*. *Proc Natl Acad Sci USA* 110(28):11320–11325.
- Layre E, et al. (2011) A comparative lipidomics platform for chemotaxonomic analysis of *Mycobacterium tuberculosis*. *Chem Biol* 18(12):1537–1549.
- Galagan JE, et al. (2013) The *Mycobacterium tuberculosis* regulatory network and hypoxia. *Nature* 499(7457):178–183.
- Madigan CA, et al. (2012) Lipidomic discovery of deoxysiderophores reveals a revised mycobactin biosynthesis pathway in *Mycobacterium tuberculosis*. *Proc Natl Acad Sci USA* 109(4):1257–1262.
- Madigan CA, et al. (2015) Lipidomic analysis links mycobactin synthase K to iron uptake and virulence in *M. tuberculosis*. *PLoS Pathog* 11(3):e1004792.
- Pacheco SA, Hsu FF, Powers KM, Purdy GE (2013) MmpL11 protein transports mycolic acid-containing lipids to the mycobacterial cell wall and contributes to biofilm formation in *Mycobacterium smegmatis*. *J Biol Chem* 288(33):24213–24222.
- Landgraf D, Okumus B, Chien P, Baker TA, Paulsson J (2012) Segregation of molecules at cell division reveals native protein localization. *Nat Methods* 9(5):480–482.
- Goedhart J, et al. (2012) Structure-guided evolution of cyan fluorescent proteins towards a quantum yield of 93%. *Nat Commun* 3:751.
- Patiño S, et al. (2008) Autofluorescence of mycobacteria as a tool for detection of *Mycobacterium tuberculosis*. *J Clin Microbiol* 46(10):3296–3302.
- Morita YS, et al. (2006) PimE is a polyprenol-phosphate-mannose-dependent mannosyltransferase that transfers the fifth mannose of phosphatidylinositol mannoside in mycobacteria. *J Biol Chem* 281(35):25143–25155.
- Christensen H, Garton NJ, Horobin RW, Minnikin DE, Barer MR (1999) Lipid domains of mycobacteria studied with fluorescent molecular probes. *Mol Microbiol* 31(5):1561–1572.
- Maloney E, et al. (2011) Alterations in phospholipid catabolism in *Mycobacterium tuberculosis* lysX mutant. *Front Microbiol* 2:19.
- Wheatley RW, Zheng RB, Richards MR, Lowary TL, Ng KK (2012) Tetrameric structure of the Glt2 galactofuranosyltransferase reveals a scaffold for the assembly of mycobacterial arabinogalactan. *J Biol Chem* 287(33):28132–28143.
- Guricha SS, et al. (2002) Ppm1, a novel polyprenol monophosphomannose synthase from *Mycobacterium tuberculosis*. *Biochem J* 365(Pt 2):441–450.
- Liu S, Neidhardt EA, Grossman TH, Ocain T, Clardy J (2000) Structures of human dihydroorotate dehydrogenase in complex with antiproliferative agents. *Structure* 8(1):25–33.
- Griffin JE, et al. (2011) High-resolution phenotypic profiling defines genes essential for mycobacterial growth and cholesterol catabolism. *PLoS Pathog* 7(9):e1002251.
- Daniel J, et al. (2004) Induction of a novel class of diacylglycerol acyltransferases and triacylglycerol accumulation in *Mycobacterium tuberculosis* as it goes into a dormancy-like state in culture. *J Bacteriol* 186(15):5017–5030.
- Guerin ME, et al. (2009) Substrate-induced conformational changes in the essential peripheral membrane-associated mannosyltransferase PimA from mycobacteria: Implications for catalysis. *J Biol Chem* 284(32):21613–21625.
- Kang CM, Nyayapathy S, Lee JY, Suh JW, Hesson RN (2008) Wag31, a homologue of the cell division protein DivIVA, regulates growth, morphology and polar cell wall synthesis in mycobacteria. *Microbiology* 154(Pt 3):725–735.

Supporting Information

Hayashi et al. 10.1073/pnas.1525165113

Movie S1:
PimE-GFP-FLAG/
HA-mCherry-Glft2

Movie S1. Time-lapse microscopy of the HA-mCherry-Glft2/PimE-GFP-FLAG dual-expression strain grown on an agar pad at 37 °C. Images were recorded for 18 h at 15-min intervals. The movie was recorded at 1,800× real-time speed.

[Movie S1](#)

Movie S2:
HA-mCherry-Glft2
post cycloserine
treatment

Movie S2. Time-lapse microscopy of the HA-mCherry-Glft2-expressing strain after DCS treatment. Representative time-lapse experiment in a constant-feed custom microfluidic device. The cells were grown in a supplemented Middlebrook 7H9 medium (*Materials and Methods*) at 30 °C. DCS at 40 µg/mL was added at 10 h and removed at 16 h. Images were recorded at 15-min intervals after removal of DCS. The movie was recorded at 5 frames per second.

[Movie S2](#)

Other Supporting Information Files

[SI Appendix \(PDF\)](#)

[Dataset S1 \(XLSX\)](#)

[Dataset S2 \(XLSX\)](#)

[Dataset S3 \(PDF\)](#)

Supplemental Information Appendix

Spatially Distinct and Metabolically Active Membrane Domain in Mycobacteria

Jennifer M. Hayashi, Chu-Yuan Luo, Jacob A. Mayfield, Tsungda Hsu,
Takeshi Fukuda, Andrew L. Walfield, Samantha R. Giffen, John Leszyk,
Christina E. Baer, Owen T. Bennion, Ashoka Madduri, Scott A. Shaffer,
Bree B. Aldridge, Christopher M. Sassetti, Steven J. Sandler, Taroh Kinoshita,
D. Branch Moody, and Yasu S. Morita

Figure Legends

Figure S1. **The protein populations of the PM-CW and the PMf are different.** The PM-CW is enriched in proteins involved in transport and metabolisms of carbohydrates, amino acids, and inorganic ions. The PM-CW also has a population of proteins involved in signal transduction, cell envelope biogenesis, and posttranslational modification. The PMf is enriched in lipid metabolism and cell envelope biogenesis. DNA replication and protein translational machinery are also enriched in the PMf, but these proteins are contaminants (see Fig. S4C).

Figure S2. **Biochemical verification of PMf-associated proteins identified by proteomic analysis.** (A) Association of HA-tagged proteins to the PMf. Lysates of *Msmeg* cells expressing HA-tagged proteins were fractionated by sucrose density gradient, and HA-tagged proteins were detected by western blotting. Protein bands were quantified to calculate % distribution. (B) PyrD-HA (38.2 kDa), an HA-tagged PMf-associated protein co-immunoprecipitates with endogenous PimB', a known PMf protein. MptA is a marker for the PM-CW. (C) Anti-HA agarose beads co-immunoprecipitate neither PimB' nor MptA from a wild-type cell lysate, showing that co-IP is dependent on the HA-tagged proteins. (D) GlnA1-HA (54.6 kDa), a non-PMf protein, does not localize to the PMf. The PMf fractions are identified by the localization of PimB' (41.4 kDa) and the PM-CW fractions are identified by the localization of MptA (54.3 kDa). (E) GlnA1-HA pulls down neither PimB' nor MptA.

Figure S3. **Immunogold labeling of *Msmeg* PMf fractions.** (A) Immunogold labeling of the PMf-containing density gradient fraction of GlfT2-HA expressing cells without primary antibodies. The samples were labeled with colloidal gold-conjugated secondary antibodies only, and quantified as

in Fig. 1D. (B) Immunogold labeling of the PMf-containing density gradient fraction of wild-type (WT) cells without primary antibodies, but with colloidal gold-conjugated secondary antibodies only. (C) Immunogold labeling of the PMf-containing density gradient fraction of wild-type cells with anti-PimB' primary antibody and colloidal gold-conjugated secondary antibody. (D) Immunogold labeling of the PMf-containing density gradient fraction of wild-type cells with anti-HA primary antibody and colloidal gold-conjugated secondary antibody. Bar = 100 nm. Closed arrowheads = 10 nm gold, anti-PimB' antibody. (E) Quantification of the wild-type colloidal gold antibody staining from Figure S3B-D. None, no gold particle detected on vesicle; HA, one or more 5 nm gold particles detected; and PimB', at least one 10 nm gold particles detected. N=60.

Figure S4. Replacement of the endogenous GlfT2 with HA-mCherry-GlfT2 for an improved PMf proteome. (A) Growth curves of HA-mCherry-GlfT2 (orange) expressing cells in comparison to wild type (green) demonstrates growth is not inhibited by the gene replacement. Growth of each strain was measured in triplicate and standard deviations are shown. (B) HA-mCherry-GlfT2 (99.6 kDa) localizes to the PMf by density gradient fractionation. The PMf, identified by PimB' (41.4 kDa); and the PM-CW, identified by MptA (54.3 kDa). (C) Protein function categorization of the refined PMf proteome generated from highly purified PMf sample by the DAVID gene functional classification tool demonstrates enrichment of biosynthetic processes including cell envelope and lipid biogenesis.

Figure S5. Ion at m/z 732.568 identified as phosphatidylethanolamine. CID spectrum of precursor ion 732.568 labeled with measured m/z values and error in parts per million (ppm) for matches to deduced fragments that match known values within 10 ppm. The maximum centroid peak intensity of a single scan is shown. The dashed red line indicates the parent ion. Dotted blue lines indicate the masses of expected fragments based on the deduced chemical structure. Inset shows the chemical structure of PE.

Figure S6. Ion at m/z 851.57 identified as phosphatidylinositol. CID spectrum of precursor ion 851.57 labeled with measured m/z values and error in parts per million (ppm) for matches to deduced fragments that match known values within 10 ppm. The maximum centroid peak intensity of a single scan is shown. The dashed red line indicates the parent ion. Dotted blue lines indicate the masses of expected fragments based on the deduced chemical structure. Inset shows the chemical structure of PI. Isomers likely exist but cannot be differentially detected by this method.

Figure S7. **Ion at m/z 1575.95 identified as AcPIM3.** CID spectrum of precursor ion 1575.95 labeled with measured m/z values and error in parts per million (ppm) for matches to deduced fragments that match known values within 10 ppm. The maximum centroid peak intensity of a single scan is shown. The dashed red line indicates the parent ion. Dotted blue lines indicate the masses of expected fragments based on the deduced chemical structure. Inset shows the chemical structure of AcPIM3. Deduced fragment masses in the inset formatted with italics were not detected at < 10 ppm but are intermediates for fragments that were detected. Isomers likely exist but were not resolved.

Figure S8. **Ion at m/z 878.82 identified as triacylglycerol.** CID spectrum of precursor ion 878.82 labeled with measured m/z values and error in parts per million (ppm) for matches to deduced fragments within 10 ppm. The maximum centroid peak intensity of a single scan is shown. The dashed red line indicates the parent ion. Dotted blue lines indicate the masses of expected fragments based on the deduced chemical structure. Inset shows the chemical structure of TAG. Isomers likely exist but were not resolved.

Figure S9. **Ion at m/z 671.466 identified as phosphatidic acid.** CID spectrum of precursor ion 671.466 labeled with measured m/z values and error in parts per million (ppm) for matches to deduced fragments that match known values within 10 ppm. The maximum centroid peak intensity of a single scan is shown. The dashed red line indicates the parent ion. Dotted blue lines indicate the masses of expected fragments based on the deduced chemical structure. Inset shows the chemical structure of PA.

Figure S10. **Ppm1 is a PMf protein and colocalizes with GlfT2 *in vitro*.** (A) Ppm1-mNG-cMyc-expressing cell lines grow similarly to wild-type *Msmeg*. Blue diamond, wild type; red square, Ppm1-mNG-cMyc expressing cells; green triangle, HA-mCherry-GlfT2 and Ppm1-mNG-cMyc dual expressing cells. Growth of each strain was measured in triplicate and standard deviations are shown. (B) Ppm1-mNG-cMyc (59.1 kDa) protein localized to the PMf containing fractions by sucrose gradient. The PMf, identified by PimB' (41.4 kDa); and the PM-CW, identified by MptA (54.3 kDa). (C) Distribution of two PMf markers (HA-mCherry-GlfT2, 99.6 kDa, and Ppm1-mNG-cMyc, 59.1 kDa) in a single cell line by sucrose gradient demonstrates fusion protein localization biochemically to the PMf-containing fractions. The PMf fractions are identified by the localization of PimB' (41.4 kDa) and the PM-CW fractions are identified by the localization of MptA (54.3 kDa). (D) Immunoprecipitation of HA-mCherry-GlfT2 from dual expressing (HA-mCherry-GlfT2, 99.6 kDa, and Ppm1-mNG-cMyc, 59.1 kDa) cell lysate pulls down the other PMf proteins, endogenous PimB' (41.4 kDa) and Ppm1-mNG-cMyc (59.1 kDa), but not the PM-CW marker MptA (54.3 kDa).

Figure S11. Fluorescent fusion proteins show the localization of the PMf and the PM-CW. (A) Biochemical analysis of the fluorescent protein localizations show that both the mTurquoise- and mCherry-labeled GlfT2 proteins (mTurQ-GlfT2-FLAG, 101.6 kDa; and HA-mCherry-GlfT2, 99.6 kDa) localize to the PMf-containing fractions. The PMf, identified by PimB' (41.4 kDa); and the PM-CW, identified by MptA (54.3 kDa). (B) Fluorescent fusion proteins of GlfT2 (mTurQ-GlfT2-FLAG and HA-mCherry-GlfT2) colocalize. (C) Distribution of a PMf-associated fusion protein (HA-mCherry-GlfT2, 99.6 kDa) and a PM-CW associated fusion protein (PimE-GFP-FLAG, 82.3 kDa) along a sucrose gradient demonstrates their different *in vitro* localizations. The PMf, identified by PimB' (41.4 kDa); and the PM-CW, identified by MptA (54.3 kDa).

Figure S12. Autofluorescence does not account for the blue fluorescence from mTurquoise-GlfT2-FLAG expressing cells. All fluorescent images were captured by 3 s exposures, and processed identically. Note that blue auto-fluorescence in wild-type cells is negligible.

Figure S13. The spatiotemporal correlation of the PMf and polar CW elongation examined by chemical labeling of CW. (A) New cell envelope growth, represented by the absence of AlexaFluor 488 amine-reactive dye (white bars), correlates with the HA-mC-GlfT2 foci (arrowheads). Bar=5µm. (B) Quantitation of AlexaFluor 488 staining and HA-mC-GlfT2 foci along the cell length. Cell images were sliced into 10 sections and average fluorescent intensities of 5 slices from the center to the pole were quantified in arbitrary fluorescent units (a.u.). n=214 cells.

Figure S14. Working model for the compartmentalization of glycerolipid biosynthesis pathways in *M. smegmatis* revealed by proteome and lipidome analyses. The data obtained from the proteome and lipidome were compiled to demonstrate one possible correlative interpretation of how the enzymes and lipids can interact with membrane compartments in mycobacteria. MSMEG_ accession numbers are indicated for known or putative enzymes involved in the pathway. Subcellular localization of proteins and lipids are color-coded as indicated. AcPIM6/lipomannan (LM)/lipoarabinomannan (LAM) biosynthesis pathways are abbreviated beyond AcPIM3 because we identified neither the enzymes nor the lipid intermediates. PI and PE are major phospholipid products. Another major phospholipid species, cardiolipin, was omitted because we did not find any enzymes involved in the cardiolipin biosynthesis. AcPIM2 and Ac₂PIM2 are well-established products of the PIM biosynthesis pathway. Ac₂PIM1 is a minor product, which is not well-characterized. G3P, glycerol-3-phosphate; PA, phosphatidic acid; DAG, diacylglycerol; TAG, triacylglycerol; FA-CoA, fatty acyl-CoA; CDP-DAG, cytidine diphosphate DAG; PPi, pyrophosphate; PI, phosphatidylinositol; PE, phosphatidylethanolamine; PS,

phosphatidylserine. GPAT, glycerol phosphate acyltransferase; AGPAT, acylglycerol phosphate acyltransferase; PAP, PA phosphatase; DAGK, DAG kinases; TGS, TAG synthetase; Psd, PS decarboxylase; PimA, PimB' and ManT, mannosyltransferases; Man AcylT, mannose acyltransferase; and Ino AcylT, inositol acyltransferase.

Figure S15. **Plasmid maps of those used in this study.** (A) Expression vectors used in this study. (B) The genomic structure of *glfT2* and *ppm1* after markerless gene replacement.

Table S1. Proteins identified in a manual scan as part of energy metabolism, lipoproteins, protein trafficking, and signal transduction in the PM–CW.

	MSMEG_	Gene Annotation		MSMEG_	Gene Annotation
Energy Metabolism	0109	NAD(P) transhydrogenase beta subunit	Signal Transduction	0028	serine/threonine-protein kinase pknB
	0110	NAD(P) transhydrogenase subunit alpha		0030	serine/threonine-protein kinases pknA
	0690	iron-sulfur-binding reductase		0033	serine/threonine phosphatase PPP
	0847	NADH-ubiquinone oxidoreductase chain 4L		0246	sensor-type histidine kinase prrB
	1654	Isocitrate dehydrogenase (NADP) lcd2		0786	serine/threonine-protein kinase pknG
	1669	succinate dehydrogenase iron-sulfur protein subunit SdhB		0963	sensor-like histidine kinase senX3
	1670	succinate dehydrogenase flavoprotein subunit		1874	two-component response regulator
	2351	Electron transfer flavoprotein beta subunit		1875	sensor histidine kinase mtrB
	2352	electron transfer flavoprotein subunit alpha		2087	mechanosensitive ion channel protein MscS
	2613	Malate:quinone oxidoreductase Mqo		3677	serine/threonine-protein kinase pknF
	3233	cytochrome ubiquinol oxidase subunit		4243	serine/threonine-protein kinase pknL
	3710	cytochrome B561		4366	serine/threonine-protein kinase pknD
	4260	cytochrome c oxidase subunit 3		4989	Two-component sensor kinase TcrY
	4261	Ubiquinol-cytochrome C reductase QcrC		5487	Two-component sensor kinase MprB
	4263	Ubiquinol-cytochrome C reductase QcrB		5663	sensor-type histidine kinase prrB
	4268	transmembrane cytochrome C oxidase (Subunit II) ctaC		5870	sensor protein
	4437	cytochrome c oxidase subunit 1		6130	two-component system sensor kinase
	4936	ATP synthase subunit beta		6880	extracellular ligand-binding receptor
	4938	ATP synthase subunit alpha	Protein Trafficking	2424	signal recognition particle-docking FtsY
	4939	ATP synthase subunit H		2960	protein translocase subunit yajC
Lipoprotein	4940	ATP synthase subunit F		2961	protein-export membrane protein secD
	4942	ATP synthase subunit a		2962	protein-export membrane protein secF
	0138	MCE-family lipoprotein LprK		3174	lipoprotein signal peptidase
	0787	glutamine-binding lipoprotein GlnH		3247	extracellular ligand-binding receptor
	0929	lipoprotein lprQ		3654	protein translocase subunit secA
	1039	FEIII-dicitrte-binding periplasmic lipoprotein FecB		3886	sec-independent protein secretion pathway component TatC
	1876	lipoprotein lpqB		4183	preprotein translocase subunit secY
	2369	lipoprotein lppZ		5872	Two component system response transcriptional positive regulator phoP
	2439	alanine rich lipoprotein lppW			
	2441	Signal peptidase I LepB			
	3851	protein LppI			
	4196	lipoprotein lppL			
	4239	protein LppM			
	4475	protein LppP			
	5005	lipoprotein LprC			
	5007	lipoprotein lprB			
	5017	lipoprotein			
	5061	sugar-binding lipoprotein LpqY			
	5416	lipoprotein LpqU			
	5429	lipoprotein lpqT			
	5513	LpqN			
	6109	lipoprotein lpqG			

Table S2. PMf-associated proteins with predicted TM domains.

Accession #	MSMEG_		# of TM
gi 399984455	0023	Hypothetical protein MSMEI_0025	1
gi 399985957	1576	Alpha/beta hydrolase fold protein	1
gi 399986320	1944	Membrane protein	2
gi 399988056	3738	GTP-binding protein engA	1
gi 399988200	3887	Sec-independent protein translocase tatA/E	1
gi 399988503	4290	Dihydroorotate dehydrogenase	1
gi 399988821	4535	Glycoside hydrolase 15-related protein	2
gi 399989349	5086	Fatty-acid-CoA ligase FadD6	1

Table S3. Summary statistics of representative lipids. The identity of at least one compound in each family was established through repeat experiments using collision-induced dissociation mass spectrometry (CID-MS, Y). Other members of the indicated family were assigned when ions showing nearly identical retention times and mass variations typical of alkyl change length or unsaturation were detected (CID-MS, N).

Lipid Class	Subclass	Formula	xcms Name	<i>m/z</i>		Deviation (ppm)	Ion	CID-MS	Fold change (log2) PMf up (+) PM-CW up (-)	Adjusted p value (Benjamini-Hochberg)	Average normalized intensity
				Expected	Observed						
glycerides	diacylated	C36H70O5	M601T179p	600.557	600.553	-6.1	[M+NH4] ⁺	N	-0.79	0.3209	16.62
		C37H70O5	M613T185p	612.557	612.557	0.35	[M+NH4] ⁺	Y	-0.08	0.9095	23.03
		C37H72O5	M615T183p	614.572	614.565	-12	[M+NH4] ⁺	N	-1.06	0.4905	19.25
		C38H74O5	M629T183p	628.588	628.585	-5.4	[M+NH4] ⁺	N	-1.87	0.0403	19.53
		C39H74O5	M641T614p	640.588	640.588	-0.67	[M+NH4] ⁺	N	0.19	0.8232	19.47
	triacylated	C53H100O6	M851T183p	850.786	850.784	-2.21	[M+NH4] ⁺	Y	1.67	0.0589	20.9
		C54H102O6	M865T184_1p	864.802	864.792	-11.66	[M+NH4] ⁺	N	1.4	0.12	17.73
		C55H104O6	M879T185p	878.818	878.815	-2.86	[M+NH4] ⁺	Y	1.96	0.0291	20.84
		C57H108O6	M907T184p	906.849	906.847	-2.53	[M+NH4] ⁺	Y	1.92	0.0862	19.84
		C59H112O6	M935T188p	934.88	934.876	-4.84	[M+NH4] ⁺	N	0.48	0.7307	17.82
		C60H116O6	M951T157p	950.911	950.903	-8.95	[M+NH4] ⁺	N	0.91	0.4081	14.81
		C61H116O6	M963T185p	962.912	962.909	-2.18	[M+NH4] ⁺	N	1.68	0.0624	20.98
		C62H120O6	M979T158p	978.943	978.933	-10.06	[M+NH4] ⁺	N	1.09	0.4436	15.68
		C63H122O6	M993T152p	992.958	992.95	-8.12	[M+NH4] ⁺	N	1.24	0.371	15.04
phosphatidic acids		C35H67O8P	M645T457n	645.45	645.45	0.37	[M-H] ⁻	Y	1.89	0.0056	19.28
		C37H69O8P	M671T454n	671.466	671.466	-0.14	[M-H] ⁻	Y	1.75	0.0218	20.1
		C37H71O8P	M673T428n	673.481	673.481	-0.34	[M-H] ⁻	Y	1.49	0.014	20.97
		C37H73O8P	M675T419n	675.497	675.491	-8.86	[M-H] ⁻	Y	2	0.0118	18.04
phosphatidyl-ethanolamine		C35H70O8NP	M662T1171n	662.477	662.477	0.6	[M-H] ⁻	N	0.21	0.8148	18.83
		C37H70O8NP	M686T1174n	686.477	686.476	-0.81	[M-H] ⁻	Y	-0.05	0.9382	18.44
		C37H72O8NP	M688T1164n	688.492	688.493	0.4	[M-H] ⁻	N	-0.2	0.6797	21.76
		C37H74O8NP	M691T1167n	690.508	690.506	-2.22	[M-H] ⁻	N	-0.75	0.2578	20.42
		C39H74O8NP	M715T1159n	714.508	714.508	0.26	[M-H] ⁻	Y	-0.07	0.9009	22.1
		C39H76O8NP	M717T1192n	716.524	716.524	0.53	[M-H] ⁻	Y	-0.03	0.9696	24.17
		C39H78O8NP	M719T1170n	718.539	718.531	-12.03	[M-H] ⁻	N	-0.51	0.3742	20.95
		C40H78O8NP	M731T1234n	730.539	730.537	-2.43	[M-H] ⁻	N	0.7	0.1868	18.94
		C40H80O8NP	M733T1200n	732.555	732.554	-1.82	[M-H] ⁻	Y	-0.53	0.5083	21.42
phosphatidylinositol		C41H79O13P	M810T1541n	809.519	809.511	-9.73	[M-H] ⁻	N	0.23	0.6617	18.79
		C42H81O13P	M824T1514n	823.534	823.538	4.22	[M-H] ⁻	N	0.28	0.5351	20.44
		C43H79O13P	M834T1533n	833.519	833.522	4.3	[M-H] ⁻	N	0.52	0.2625	22.28
		C43H81O13P	M836T1515n	835.534	835.539	5.2	[M-H] ⁻	N	0.17	0.7265	25.83
		C43H83O13P	M838T1514n	837.55	837.543	-8.62	[M-H] ⁻	N	0.26	0.5546	22.9
		C44H83O13P	M850T1500n	849.55	849.553	4.13	[M-H] ⁻	N	0.63	0.1681	22.05
		C44H85O13P	M852T1484n	851.566	851.569	4.67	[M-H] ⁻	Y	0.22	0.5938	24.99
		C45H85O13P	M864T1484n	863.566	863.566	0.13	[M-H] ⁻	N	0.12	0.7979	20.39
		C45H87O13P	M866T1482n	865.581	865.576	-6.23	[M-H] ⁻	N	0.64	0.2327	18.39
		C46H87O13P	M878T1477n	877.581	877.581	0.02	[M-H] ⁻	N	0.97	0.103	17.38
		C46H89O13P	M880T1463n	879.597	879.597	0.33	[M-H] ⁻	N	0.51	0.3363	18.7
phosphatidylinositol monomannosides	diacyl	C82H155O20P	M1492T1504p	1492.092	1492.09	-1.42	[M+H] ⁺	Y	-1.31	0.0333	15.36
phosphatidylinositol dimannoside	monoacyl	C72H135O24P	M1414T1720n	1413.901	1413.901	-0.08	[M-H] ⁻	N	-0.84	0.2176	23.1
		C74H139O24P	M1442T1696n	1441.932	1441.931	-0.57	[M-H] ⁻	N	-0.32	0.6053	17.75
	diacyl	C88H165O25P	M1671T1501p	1671.172	1671.164	-4.52	[M+NH4] ⁺	Y	-1.2	0.0436	19.01
		C90H169O25P	M1680T1470n	1680.162	1680.156	-3.13	[M-H] ⁻	N	-0.78	0.1386	18.38
		C91H171O25P	M1713T1469p	1713.219	1713.21	-4.88	[M+NH4] ⁺	Y	-2.32	0.0367	16.86
phosphatidylinositol trimannoside	monoacyl	C78H145O29P	M1576T1991n	1575.954	1575.946	-4.77	[M-H] ⁻	Y	-1.86	0.0068	18.96

Table S4. **CID fragmentation of alkyl series.** Parent and CID fragments detected by mass spectrometry for ions in addition to those profiled in Figure 2 & Figures S5–S9.

Lipid Class	Subclass	Formula	Parent <i>mz</i>	Collision energy (mV)	Retention time (s)	Fragment formula	Fragment <i>mz</i>		Deviation (ppm)	Intensity
							Expected	Observed		
glycerides	diacylated	C37H70O5	612.557	25	175.44	C19H37O3+	313.274	313.273	1.18	5136.29
						C21H39O3+	339.289	339.289	0.37	2917.70
						C21H41O3+	341.305	341.304	3.56	54.13
						C37H69O4+	577.519	577.517	3.63	1277.17
	triacylated	C53H100O6	850.786	25	193.38	C35H65O4+	549.488	549.485	5.09	96.90
						C35H67O4+	551.503	551.502	2.40	463.09
		C54H102O6	864.802	25	169.5	C36H69O4+	565.519	565.516	4.96	175.78
						C38H71O4+	591.535	591.535	0.11	114.78
						C38H73O4+	593.550	593.550	1.15	27.08
		C57H108O6	906.849	25	172.5	C37H69O4+	577.519	577.521	3.97	325.69
						C39H73O4+	605.550	605.546	7.04	103.01
						C39H75O4+	607.566	607.570	6.74	580.44
						C41H77O4+	633.582	633.577	7.63	273.00
phosphatidic acids		C35H67O8P	645.45	25	428.76	C3H6O5P-	152.996	152.996	2.52	896.99
						C14H27O2-	227.202	227.203	5.62	480.09
						C16H29O2-	253.217	253.217	0.00	132.08
						C18H33O2-	281.249	281.250	5.56	1377.39
		C37H71O8P	673.481	35	433.92	C3H6O5P-	152.996	152.996	1.57	664.12
						C16H29O2-	253.217	253.220	8.97	98.01
						C16H31O2-	255.233	255.233	0.99	1170.56
						C18H33O2-	281.249	281.249	0.41	3156.61
		C37H73O8P	675.497	25	509.22	C3H6O5P-	152.996	152.995	3.36	97.90
						C16H31O2-	255.233	255.233	1.59	115.41
						C18H33O2-	281.249	281.249	2.69	56.01
						C18H35O2-	283.264	283.265	2.62	300.74
phosphatidyl-ethanolamine		C37H70NO8P	686.488	35	1287.48	C5H11NO5P-	196.038	196.036	8.55	134.08
						C14H27O2-	227.202	227.207	21.40	25.50
						C16H29O2-	253.217	253.220	12.08	147.56
						C18H33O2-	281.249	281.252	10.22	156.95
		C39H74NO8P	714.515	45	1452.66	C5H11NO5P-	196.038	196.041	13.89	15.44
						C16H29O2-	253.217	253.221	12.74	173.01
		C39H76NO8P	716.538	35	1965.54	C5H11NO5P-	196.038	196.040	10.60	72.07
						C16H29O2-	253.217	253.215	10.34	11.01
						C16H31O2-	255.233	255.235	9.75	608.89
						C18H33O2-	281.249	281.247	4.78	289.00
						C19H37O2-	297.280	297.285	15.33	48.08
phosphatidylinositol monomannosides	diacyl	C82H155O20P	1492.092	45	1455.9	C38H73O4+	593.550	593.550	1.40	46.60
						C82H155O20P+	1492.092	1492.088	2.53	6.16
phosphatidylinositol dimannoside	diacyl	C88H165O25P	1671.172	45	1438.14	C35H67O4+	551.503	551.507	6.53	10.03
						C38H73O4+	593.550	593.549	1.92	2222.64
						C88H169NO25P+	1671.172	1671.176	2.39	14.67
	diacyl	C91H171O25P	1713.219	45	1646.34	C38H73O4+	593.550	593.556	9.47	53.53
						C91H175NO25P+	1713.219	1713.228	5.55	6.00

Table S5. **Primers to create plasmids used in this study.** Key restriction enzyme sites are underlined.

Plasmid	Gene ID	Protein	Forward Primer Reverse Primer
pYAB279	MSMEG_0389	Gtf1	ATTAAATTTGTGCTGGCGAGTTACG ATTTCGCCGAACACCTCGACCT
pYAB280	MSMEG_2308	Geranylgeranyl reductase	ATTCTGACCTGCTCAGACCGC ATTCTTCTTGATGACGTGCTGCG
pYAB281	MSMEG_6403	GltF2	ATTAGTGACATCCCTTCCGGC ATTTTCGTCCGACTTTCTCCGGT
pMUM012	MSMEG_4290	GlnA1	ATTGATATCGTGGCAGAAAAGACGTCCGA ATTAGTACTCACGTCGTAGTAGAGCGAGAACT
pMUM015	MSMEG_4290	GlnA1	GTGGCAGAAAAGACGTCCGA CACGTCGTAGTAGAGCGAGAACT
pMUM023	MSMEG_4198	PyrD	ATCCATATGTACCGCGTTCTGCGG ACTGCGCACGGCTGATCCGACA
pMUM024	MSMEG_1944	Membrane Protein	ATCCATATGCCTGGACCGATCCGAG ACTGGAGTGGCGGTACACCTTGA
pMUM043	mTurquoise	mTurquoise- GltF2-FLAG	GTACATCAATTGTCATGACAGTTAGTAAAAGAGAAGAACTTTT CAC
	FLAG		TGATATCAATTGCAGTACTTTTGTAGAGCTCATCCATGCCA GTGTCCAAGCCCGCCAT
			GTCGACATCGATTTCATTATTTGTCGTCGTCGTCTTTGTAGTCC TCGAGAATATTTTCGTCCGACTTTCTCCG
			TTTTTTTTTTCAGAGACTGCGACGCAGTTGGGTAAACGGGTCCT TTTTTTTTTTCAGCATCTGTTGCTGCTTTTCGTCTTCGGTGA TACCCGTACGACGTCCCGGACTACGCGGCG CGCGTAGTCCGGGACGTCGTACGGGTACAT TTTTTTTTTTCACGCGGTGAAGTTTCCAAGGGCGAGGAGGATAA CAT TTTTTTTTTTCACGCGGTGTTTGTAGAGCTCATCCATGCCACCG TTTTTTTTTTCACCGCGTGAGTGACATCCCTTCCGGCGCACTC TTTTTTTTTTCACCTTGTGTCATGCACATCCACCAGCCGTTGAA
pMUM053	Upstream of MSMEG_6403	GltF2 gene replacement	TTTTTTTTTTCAGAGACTGACGAGATATCGCGCACCAG TTTTTTTTTTCAGATTCTGATTAATGCGGACCACGCCCTG
	HA		AAAACCCCATTAAATGTTAGTAAAAGAGAAGAAGATAA TTTTCCCCCTTAAGCAAGTCTTCTTCCGAGATCAGCTTTTGT CAGTACTTTTGTAGAGCTCATCCAT
	mCherry		TTTTTTTTTTCAGCAGGTGTTTGTAGAGCTCATCCATGCCACCG TTTTTTTTTTCACCGCGTGAGTGACATCCCTTCCGGCGCACTC TTTTTTTTTTCACCTTGTGTCATGCACATCCACCAGCCGTTGAA
	Downstream of MSMEG_6403		TTTTTTTTTTCAGAGACTGACGAGATATCGCGCACCAG TTTTTTTTTTCAGATTCTGATTAATGCGGACCACGCCCTG
pMUM072	Upstream of MSMEG_3859	Ppm1 gene replacement	AAAACCCCATTAAATGTTAGTAAAAGAGAAGAAGATAA TTTTCCCCCTTAAGCAAGTCTTCTTCCGAGATCAGCTTTTGT CAGTACTTTTGTAGAGCTCATCCAT
	mNeonGreen		TTTTTTTTTTCAGAATCTGCTTAAGTGACGGGTAGCGTCCG TTTTTTTTTTCAGCTTCTGGTGTCCAACAC GAAATCATCG
			Downstream of MSMEG_3859

Dataset Legends

Dataset S1. List of proteins associated with PMf, PM-CW, or refined PMf identified by proteomic analysis. This dataset is provided as a separate Excel separate file. First tab, PM-CW; second tab, PMf; and third tab, refined PMf.

Dataset S2. The quadruplicate analysis of the PMf and PM-CW lipidomes. This dataset is provided as a separate Excel file.

Dataset S3. R script for the lipidomics analysis. A script in .R format for all aspects of the lipidomics analysis including ion peak picking and statistical analysis. Dataset S2, Figure 2 and Figures S5–9 were produced using this script. Raw MS data are available by request.

Materials and Methods

Construction of plasmids

Main features of each plasmid are summarized in Fig. S15.

pYAB186 — To create an expression vector for PimE-GFP-FLAG (*pYAB186*), *pYAB052* (1) was digested with *Ssp*I, and *pEGFP-C1* (Clontech Laboratories) was digested with *Eco*RI and *Age*I. The EGFP fragment was then blunt ended and ligated into the *pYAB052* fragment.

pYAB279, *pYAB280*, *pYAB281*, and *pMUM015* — To create expression vectors for MSMEG_0389 (*pYAB279*), MSMEG_2308 (*pYAB280*), MSMEG_6403 (*pYAB281*), and *GlnA1* (*pMUM015*), genes were amplified by PCR (Table S5), and the PCR products were then inserted by blunt-end ligation to *pYAB178* after removing *pimB'* gene by *Ssp*I. *pYAB178* is a hygromycin-resistant integrative *PimB'*-HA expression vector (not used in this study), which is based on *PMV361* (2). It was designed so that *pimB'* gene can be excised from *pYAB178* in frame by *Ssp*I, a blunt-end 6-bp cutter. The digested fragment can then be used to insert other genes by blunt-end ligation to create an expression vector for C-terminally HA epitope-tagged fusion proteins.

pMUM012 — *pMUM012* was used only to clone genes encoding *PyrD* and MSMEG_1944 (*pMUM023* and *pMUM024*, see below). To create, *pMUM011*, a streptomycin-resistant derivative of *pYAB178* (see above), was digested with *Ssp*I to remove the *pimB'* gene, and the PCR product 5'-*Ssp*I-*Eco*RV-*glnA1*-*Scal*-*Ssp*I-3' (restriction enzyme sites underlined in Table S5) was inserted by blunt-end ligation.

pMUM023 and *pMUM024* — The genes encoding *PyrD* (*pMUM023*) and MSMEG_1944 (*pMUM024*) were amplified by PCR (Table S5). *pMUM012* was digested with *Eco*RV and *Scal* to remove an inserted gene in frame, and the PCR fragments carrying *pyrD* and *msmeg_1944* genes were inserted by blunt-end ligation, creating expression vectors for C-terminally HA epitope-tagged fusion proteins.

pMUM043 — To construct the mTurquoise-Glft2-FLAG expression vector, the mTurquoise gene was amplified from an *Escherichia coli* strain carrying mTurquoise using the primers (Table S5). The PCR product was digested with *Mfe*I and inserted into the *Mfe*I site of *pYAB281*. We then amplified the 3' region of *Glft2* from *pYAB281* using the primers (Table S5) to add the FLAG tag in frame with the *Glft2* coding region. This PCR fragment as well as mTurquoise-inserted *pYAB281* were digested with *Cla*I and *Pml*II and ligated together to create *pMUM043*.

pMUM053 — To replace the endogenous *glft2* gene with a gene encoding HA-mCherry-Glft2, we amplified upstream and downstream regions of *glft2* using the primers shown in Table S5, and digested with *Alw*NI and *Dra*III, respectively. We also amplified mCherry using the primers shown in Table S5, and digested with *Dra*III. HA epitope was synthesized with appropriate sticky ends (Table S5). All 4 fragments were ligated into *pCOM1* (Tiwari S and Jacobs WR Jr.,

unpublished) at Van91I sites. The resulting plasmid, pMUM053, was transformed into *Msmeg* by electroporation, and positive clones were isolated using hygromycin resistance marker and SacB-dependent sucrose sensitivity. Correct replacement of the *glfT2* gene was confirmed by PCR.

pMUM072 — To replace the endogenous *ppm1* gene with a gene encoding Ppm1-mNeonGreen-cMyc, we amplified upstream and downstream regions of *ppm1* using the primers shown in Table S5, and digested with AlwNI. Both fragments and Van91I-digested pCOM1 fragment were ligated, resulting in pMUM070. *VspI* and *BspTI* sites introduced to the primers to amplify the upstream and downstream regions of *ppm1* (underlined in Table S5) were used to linearize pMUM070. We then amplified mNeonGreen-cMyc with *VspI* and *BspTI* at 5' and 3' ends, respectively (Table S5), digested with *VspI* and *BspTI*, and ligated into the pMUM070 fragment. The resultant plasmid pMUM072 was transformed into *Msmeg* by electroporation, and positive clones were selected as described for pMUM053 (see above).

Cell culture and transformation

M. smegmatis mc²155 was grown at 30°C in Middlebrook 7H9 broth supplemented with 11 mM glucose, 14.5mM NaCl, and 0.05% Tween-80, or at 37°C on Middlebrook 7H10 agar supplemented with 11 mM glucose and 14.5mM NaCl, unless otherwise indicated. Plasmid constructs (*SI Appendix*, Fig. S15) were electroporated into *Msmeg* cells as before (3). Antibiotic concentrations used were 100µg/ml hygromycin (Wako) and 50µg/ml streptomycin (Fisher Scientific).

Density gradient fractionation

Log phase cells (OD₆₀₀ = 0.5–1.0) were pelleted, washed in 50 mM Hepes (pH 7.4) buffer and resuspended in lysis buffer (25 mM Hepes, pH 7.4, 20% Sucrose, 2 mM EGTA, and a protease inhibitor cocktail) at 1 g wet pellet to 5 ml lysis buffer. Cell suspension was then subjected to nitrogen cavitation at ~2250 psi for 30min three times. Lysate was spun at 3220x *g* for 10 min at 4°C twice before loaded on a 20–50% sucrose gradient. Gradients were spun at 218,000x *g* for 6 h at 4°C, fractionated into 1 ml fractions, and used for further biochemical analysis.

Proteome preparation and analysis

PMf and PM-CW proteomics samples were prepared by pelleting pooled sucrose density gradient fractions (PMf: 4–6, PM-CW: 8–10) at 100,000x *g* and washed with HES buffer (25mM HEPES pH7.4, 2mM EGTA, 150mM NaCl). Protein samples were separated on SDS-PAGE (12% gel) for a short distance and a gel slice containing entire unresolved protein band was digested with trypsin or chymotrypsin. The extracted peptide fragments were then analyzed by data-dependent nano-LC ESI MS on an Orbitrap mass spectrometer (Thermo Scientific Q Exactive).

The data were batch searched against the *Msmeg* proteome database, using the Mascot search engine (Matrix Science, London, UK; version 2.4.0). We searched the Msmegmatis_81012 database (6690 entries) with a parent ion mass tolerance of 10.0 ppm and a fragment ion mass tolerance of 0.050 Da. Carbamidomethyl of cysteine was specified as a fixed modification. Pyro glutamic for N-terminal glutamine, oxidation of methionine and acetyl of the protein N-terminus were specified as variable modifications. Scaffold (version 4.3.2, Proteome Software Inc., Portland, OR) was then used to validate MS/MS based peptide and protein identifications. Peptide identifications were accepted if they could be established at greater than 92.0% probability by the Peptide Prophet algorithm (4). Protein identifications were accepted if they could be established at greater than 90.0% probability and contained at least 2 identified peptides. Protein probabilities were assigned by the Protein Prophet algorithm (5). Proteins that contained similar peptides and could not be differentiated based on MS/MS analysis alone were grouped to satisfy the principles of parsimony. To calculate relative protein abundances, the exponentially modified protein abundance index (emPAI) was used in the Scaffold program (6). Identified proteins with minimum spectral counts of 10 and a twofold or greater enrichment in either the PMf or the PM-CW were deemed to be part of the respective proteome.

Protein identification numbers were input into the UniProt Retrieve/ID Mapping function, the UniProt Identification numbers retrieved, and subsequently subjected to DAVID Bioinformatics Functional Annotation tool to be classified into GO Ontology groups. To select the PMf-associated proteins for further biochemical analysis, we used several independent criteria such as known or predicted function in cell envelope biogenesis, their abundance (MS spectral counts, >20), protein size (<50 kDa), and the presence of a TM domain. GlfT2 (MSMEG_6403, UDP-galactosyl transferase) was chosen for the abundance (spectral counts of 149 from trypsin digests) and known function; Gtf1 (MSMEG_0389, glycosyltransferase) and a geranylgeranyl reductase (MSMEG_2308) are chosen for their predicted function, abundance (spectral counts of 21 and 183 from trypsin digests), and a relatively small size (46.5 and 43.1 kDa respectively); PyrD is a quinone-dependent dehydrogenase and was chosen for abundance (spectral counts of 50 from trypsin digests), a relatively small size (37.2 kDa), and the presence of predicted TM domain; and MSMEG_1944 was chosen despite the fact that it has no known functions because it is a unique protein in having a particularly small size (13.9 kDa) with two predicted TM domains.

SDS-PAGE and western blotting

Protein samples were mixed with a reducing sample loading buffer, denatured on ice, and run in a 12% SDS-PAGE gel. Proteins were transferred to a PVDF membrane (BioRad), blocked in 5% milk in PBST20 (PBS with 0.05% Tween 20). Membranes were incubated with primary antibodies (anti-HA (Sigma), anti-PimB' (7), and anti-MptA (7)) at 1:2000 dilutions, washed in PBST20, and incubated

with horseradish peroxidase–conjugated secondary antibodies (Sigma) at 1:2000 dilutions. Blots were washed again in PBST20 and developed for chemiluminescence. Images were recorded using an ImageQuant LAS 4000mini (GE Healthcare). For western blotting of sucrose gradients, an equal volume of each fraction was loaded into the gel. All experiments were repeated at least twice. The tagged protein in each fraction was quantified from western blot images by ImageJ. The amount of tagged protein detected in the major gradient categories (Cytoplasm, Fr. 1–2; PMf, Fr.4–6; PM–CW, Fr. 9–11) was summed and a percentage of detected tagged protein was calculated by category.

Immunoprecipitation of PMf proteins in membrane fractions

Cell lysate (3.6 mg wet pellet equivalent in 18 μ l) was diluted 1:20 in HES or HESD (HES plus 1% Triton X–100 and 1% sodium deoxycholate) buffer, and mixed with 2.5% total volume of anti–HA agarose beads (Sigma) pre–equilibrated with either HES or HESD. After overnight incubation at 4°C, beads were washed with the respective buffers, and bound proteins were eluted by 1 mg/ml HA peptides (AnaSpec Inc) at 30°C for 15 min, twice. SDS–PAGE and western blotting were performed as before (7). All experiments were repeated at least twice to confirm the reproducibility.

Negative staining EM

Formvar–coated grids (Ted Pella) were incubated with sample (the PMf fraction from various strains as indicated in the Figure Legends) for 5 min, washed with dilution buffer (PBS with 2% BSA), incubated with primary antibody (anti–HA and/or anti–PimB') for 30–60 min, washed in buffer, and incubated with colloidal gold–conjugated secondary antibody (Ted Pella) for 30 min. Finally, grids were washed in buffer, then water, and stained with 2% uranyl acetate for 2 min. Grids were desiccated for 3–5 d. Samples on the grids were observed at 15–25,000x using a JOEL JEM–2000FX Transmission Electron Microscope with a LaB₆ electron source. A vesicle was defined as an oval shape having a minimum short axis of 38 nm and a maximum long axis of 85 nm. Most structures were circular with an average diameter of 55 nm. Sixty vesicles from each grid were quantified for their antibody binding. All experiments were repeated at least twice to confirm the reproducibility.

Refined PMf proteome preparation and analysis

The PMf fractions from cells expressing HA–mCherry–GltT2 or from wild type cells were pooled and applied to a 10– μ l bed volume of washed anti–HA agarose beads (Sigma) on a rotating wheel overnight at 4 °C as described above. Samples were then run in a 12% acrylamide gel for a short

distance, entire unresolved protein band cut out, digested with trypsin, and analyzed by LC-ESI MS as described previously.

The proteome data were analyzed as above except that the Scaffold (version 4.4.5) was used and peptide identification threshold was set at 80.0% probability. To calculate relative protein abundances, the intensity-based absolute quantification (iBAQ) was used in the Scaffold program. Identified proteins with minimum quantitative iBAQ value of 2.0×10^7 and a 10-fold or greater enrichment in the transgenic PMf sample over that of wild type were deemed to be part of the PMf proteome. The protein identification numbers were submitted to UniProt Retrieval/ID Mapping, and the list of retrieved UniProt numbers was subjected to the DAVID Bioinformatics software Functional Annotation program as above.

HPLC-MS lipidomic analysis

Sucrose density gradient fractions containing either PMf or PM-CW were generated from wild-type lysate, and subjected to chloroform/methanol/water (8:4:3) partition. The lower organic phase was washed twice with chloroform/methanol/water (3:48:47). Ten percent of the volume was removed for quantification by orcinol staining of TLC plates using Fiji (8), and the final organic phase was dried under nitrogen. This was repeated for four independent samples, with the PMf and PM-CW fraction for each sample treated as a pair.

The dried samples were resolvated at a concentration of 1 mg lipid per milliliter in hexanes/isopropanol (7:3) with 0.1% (v/v) formic acid and 0.05% (v/v) ammonium hydroxide for lipidomic profiling. Samples were centrifuged at 2,000 rpm for 10 min prior to transfer to autosampler vials (Agilent). HPLC separation and MS were performed as described (9) using an Agilent Technologies 6530 Accurate-Mass Q-TOF with a monochrome diol column. Twenty μ l volumes of samples were injected in randomized order. Each set of injections was repeated in randomized blocks in positive and negative ion mode two times, with additional injections performed using CID-scanning mode. Additional CID-MS data were collected with the same samples at energies between 25–65 V.

Raw profile mode data were converted to centroid mode and mzData format using MassHunter (Agilent). Ion peaks were identified, grouped and aligned using the R package XCMS (10) treating positive and negative mode data separately. Zero values were replaced using random minima and samples were normalized by full quantiles. Technical replicates were averaged and positive and negative ion mode data joined for statistical analysis. The R package limma (11) was used to identify ions meeting criteria for differential expression between the PMf and PM-CW, treating the PMf and PM-CW fractions for each of the four samples as a pair. Ions with an adjusted p value < 0.05 (12) and a fold-change > 2 were considered significantly different. A complete R script for analysis including all XCMS and limma parameters is available as supplemental

information (Datasets S2–S3). Ions were identified by matching mass and retention time to the MycoMap and MycoMass databases (9) followed by CID–MS.

Fluorescent microscopy

Msmeg cells expressing fluorescent proteins were grown in Middlebrook 7H9 media to logarithmic phase ($OD_{600} = 0.5\text{--}1.0$), 2 μl of cells were spotted onto a 1% agar pad made from Middlebrook 7H9 medium, and covered with a glass coverslip. All static live cell images were taken under identical settings (100x objective and 175 ms exposure for phase contrast or 3 s exposure for fluorescence), using a Nikon ECLIPSE E600 microscope equipped with ORCA–ER cooled CCD camera (Hamamatsu) and Openlab software 5.5.2 (Improvision). Pearson co-localization coefficient was calculated using Fiji (8), and average \pm standard deviation ($n=20$) is indicated.

Time-lapse microscopy

Standard time-lapse imaging was performed as before (13). Briefly, bacteria were grown on an agar pad and images were taken every 15 min for a duration of 18 h using a DeltaVision Personal DV microscope equipped with an environmental chamber warmed at 37°C (Applied Precision).

Cell wall labeling

Msmeg cells were grown to logarithmic phase ($OD_{600} = 0.5\text{--}1.0$), pelleted, washed with PBS, labeled with amine-reactive Alexa Fluor 488 succinimidyl esters (Invitrogen) at a final concentration of 0.05 mg/ml, pelleted, washed with PBS and replaced into fresh media. Cells were allowed to grow for 2.5 h at 30°C, shaking, and imaged as above. Newly elongated CW was differentiated from preexisting CW by chemical staining as before (14) and intensity profiles of fluorescent probes were analyzed by contouring randomly selected 214 cells using MicrobeTracker (15), and quantifying using MATLAB R2012a (Mathworks). Statistical significance was determined by Student's t-test.

Induction of ectopic growth poles

For DCS treatment, cells were grown to logarithmic phase and then loaded into a custom-built constant-feed microfluidic device (14) for live cell imaging. A DeltaVision PersonalDV widefield fluorescence microscope with solid state illumination and hardware-based autofocus was used to image cells every 15 min in a controlled environmental chamber warmed to 30°C. Growth medium (Middlebrook 7H9 supplemented with 0.05% Tween–80 and 10% ADC) was passed through the microfluidic device for 10 h before treatment for 6 h with 40 $\mu\text{g}/\text{ml}$ of DCS. Antibiotic treatment was followed by an 8 h recovery in regular medium.

References

1. Morita YS, *et al.* (2006) PimE is a polyprenol–phosphate–mannose–dependent mannosyltransferase that transfers the fifth mannose of phosphatidylinositol mannoside in mycobacteria. *J Biol Chem* 281:25143–25155.
2. Stover CK, *et al.* (1991) New use of BCG for recombinant vaccines. *Nature* 351:456–460.
3. Fukuda T, *et al.* (2013) Critical roles for lipomannan and lipoarabinomannan in cell wall integrity of mycobacteria and pathogenesis of tuberculosis. *mBio* 4:e00472–00412.
4. Keller A, Nesvizhskii AI, Kolker E, & Aebersold R (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. *Anal Chem* 74:5383–5392.
5. Nesvizhskii AI, Keller A, Kolker E, & Aebersold R (2003) A statistical model for identifying proteins by tandem mass spectrometry. *Anal Chem* 75:4646–4658.
6. Ishihama Y, *et al.* (2005) Exponentially modified protein abundance index (emPAI) for estimation of absolute protein amount in proteomics by the number of sequenced peptides per protein. *Mol Cell Proteomics* 4:1265–1272.
7. Sena CB, *et al.* (2010) Controlled expression of branch–forming mannosyltransferase is critical for mycobacterial lipoarabinomannan biosynthesis. *J Biol Chem* 285:13326–13336.
8. Schindelin J, *et al.* (2012) Fiji: an open–source platform for biological–image analysis. *Nat Methods* 9:676–682.
9. Layre E, *et al.* (2011) A comparative lipidomics platform for chemotaxonomic analysis of *Mycobacterium tuberculosis*. *Chem Biol* 18:1537–1549.
10. Smith CA, Want EJ, O'Maille G, Abagyan R, & Siuzdak G (2006) XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Anal Chem* 78:779–787.
11. Ritchie ME, *et al.* (2015) limma powers differential expression analyses for RNA–sequencing and microarray studies. *Nucleic Acids Res* 43:e47.
12. Benjamini Y & Hochberg Y (1995) Controlling the False Discovery Rate – a Practical and Powerful Approach to Multiple Testing. *J Roy Stat Soc B Met* 57:289–300.
13. Meniche X, *et al.* (2014) Subpolar addition of new cell wall is directed by DivIVA in mycobacteria. *Proc Natl Acad Sci U S A* 111:E3243–3251.
14. Aldridge BB, *et al.* (2012) Asymmetry and aging of mycobacterial cells lead to variable growth and antibiotic susceptibility. *Science* 335:100–104.
15. Sliusarenko O, Heinritz J, Emonet T, & Jacobs–Wagner C (2011) High–throughput, subpixel precision analysis of bacterial morphogenesis and intracellular spatio–temporal dynamics. *Mol Microbiol* 80:612–627.

Figure S1

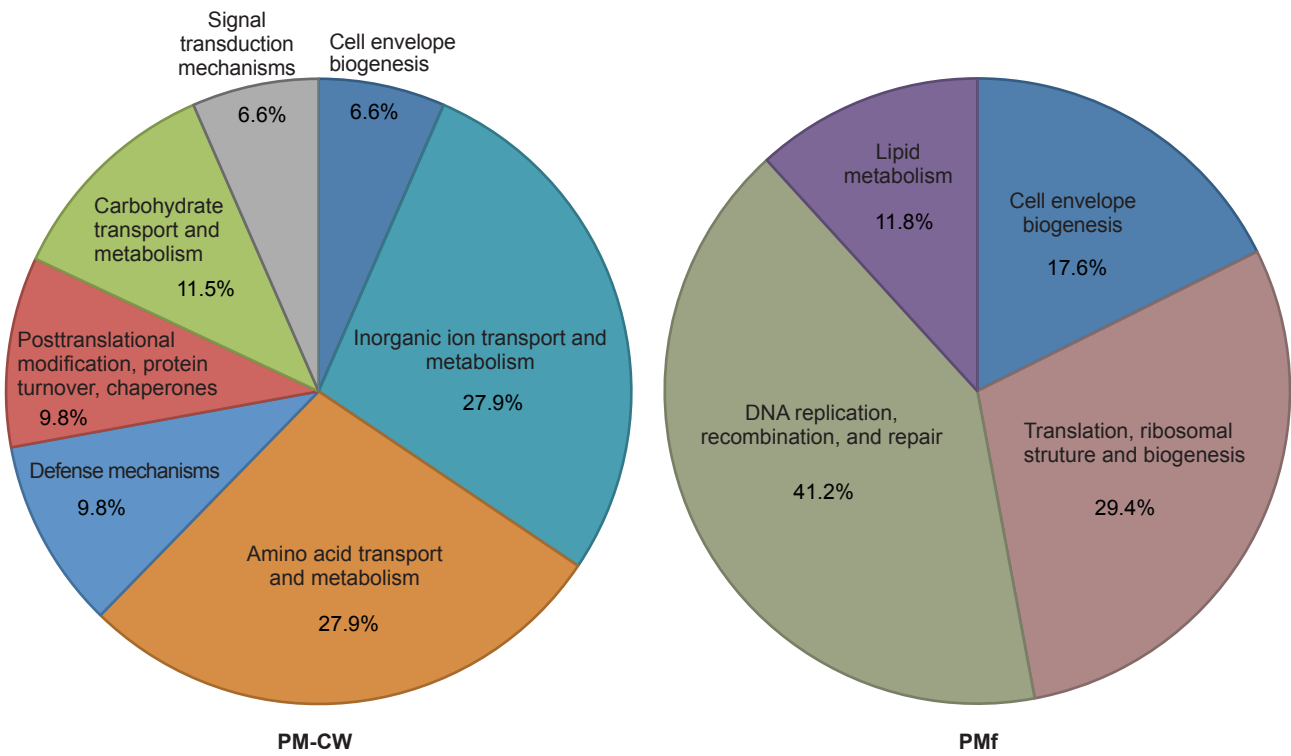


Figure S2

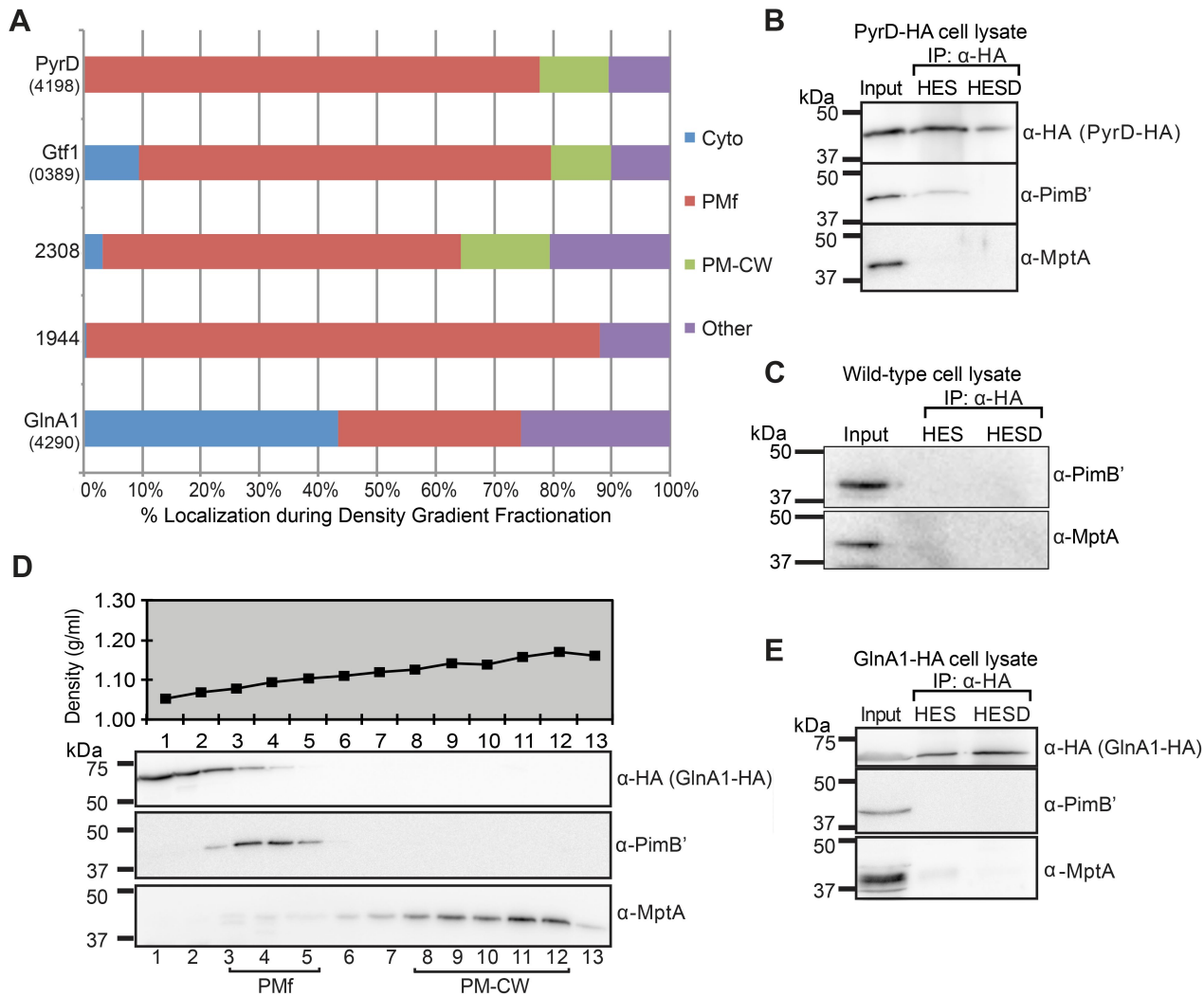


Figure S3

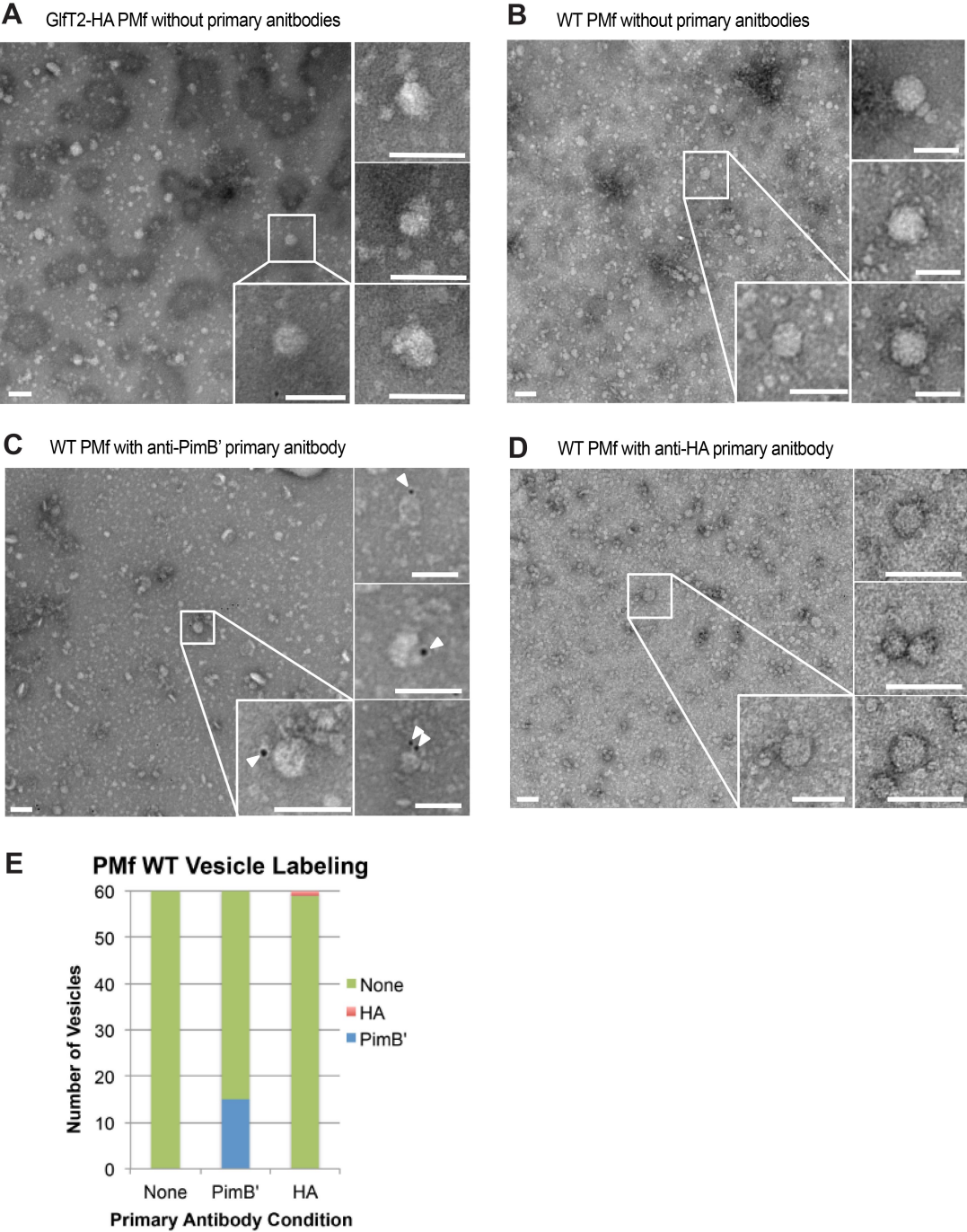


Figure S4

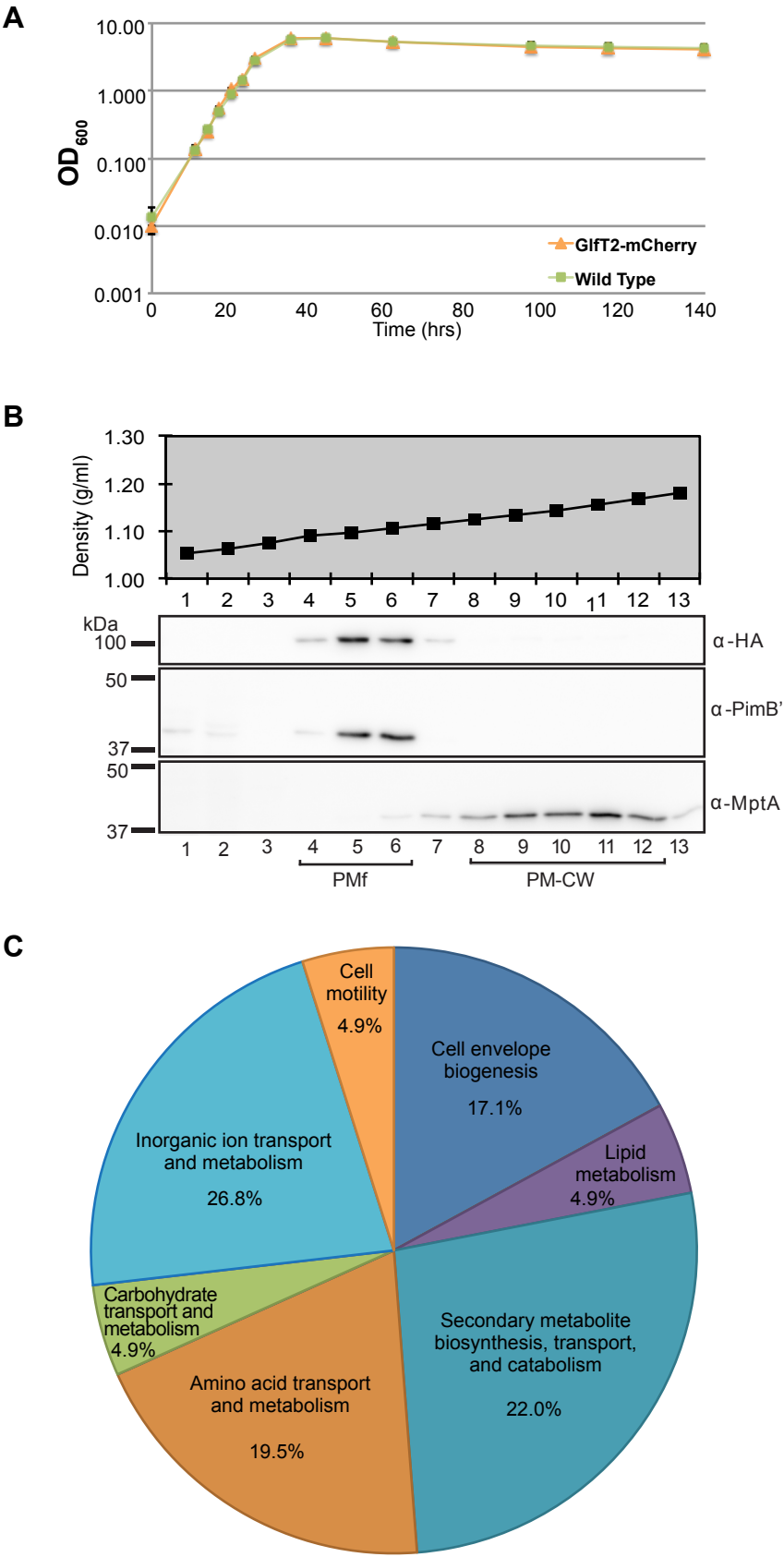


Figure S5

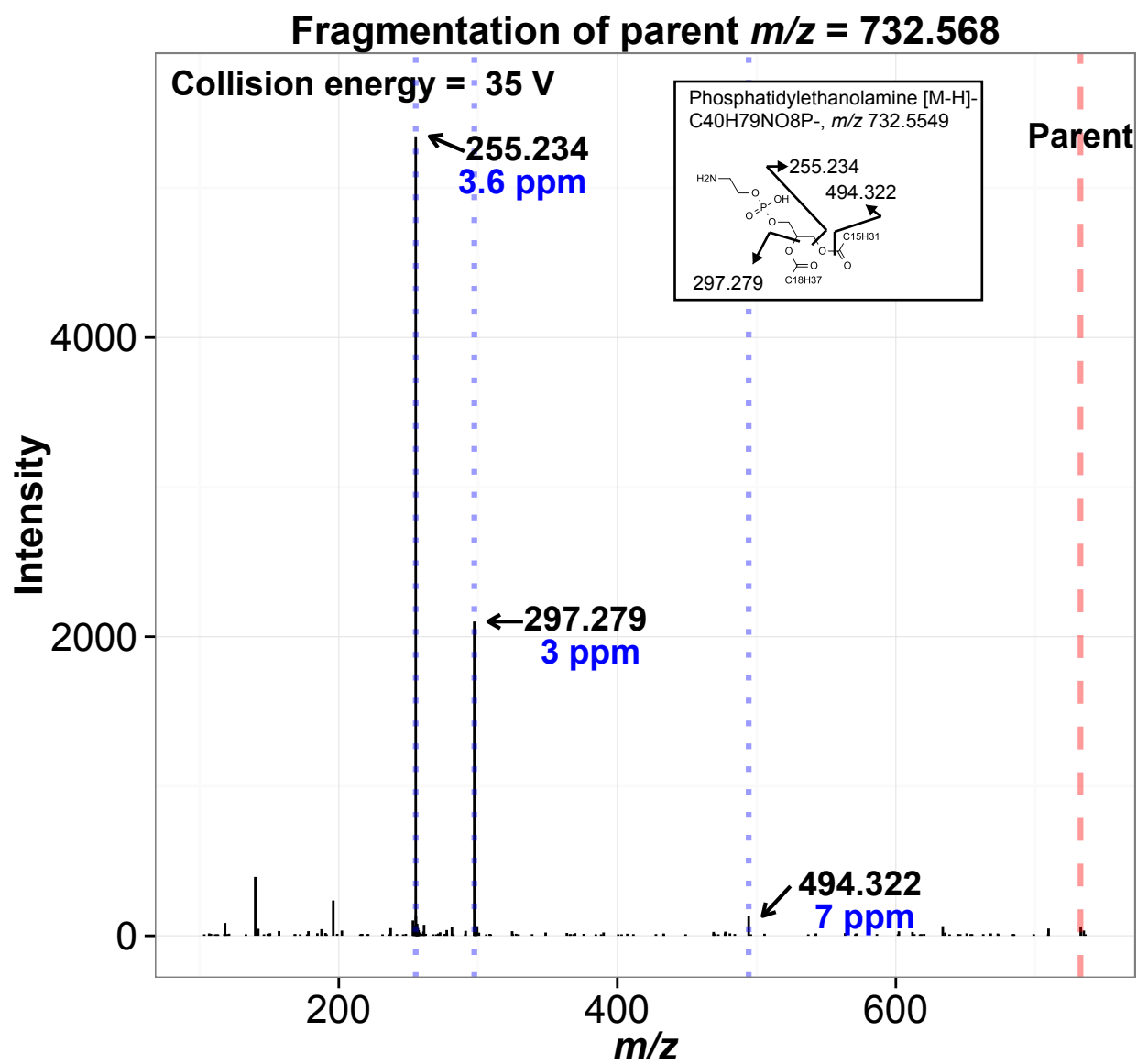


Figure S6

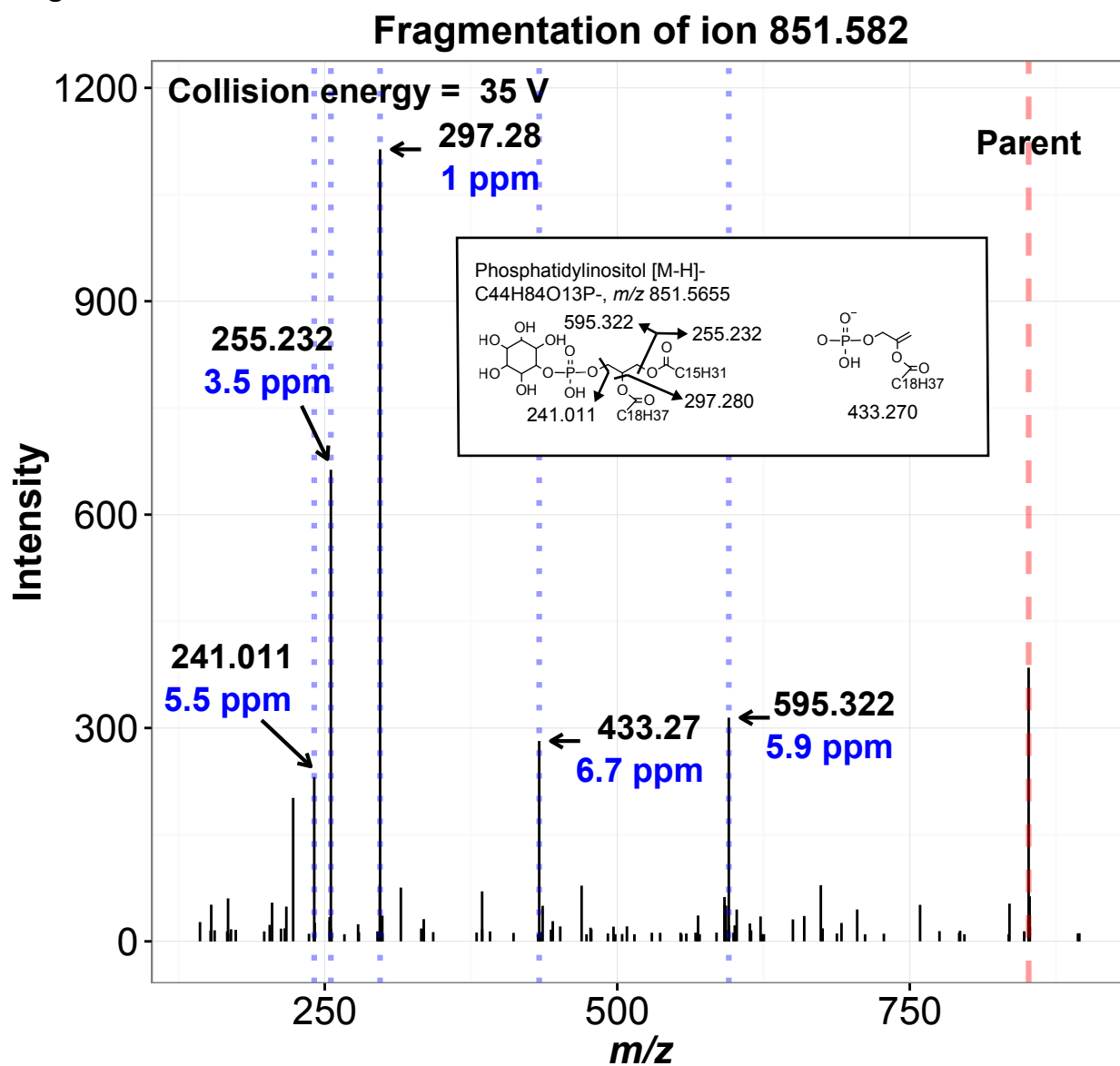


Figure S7

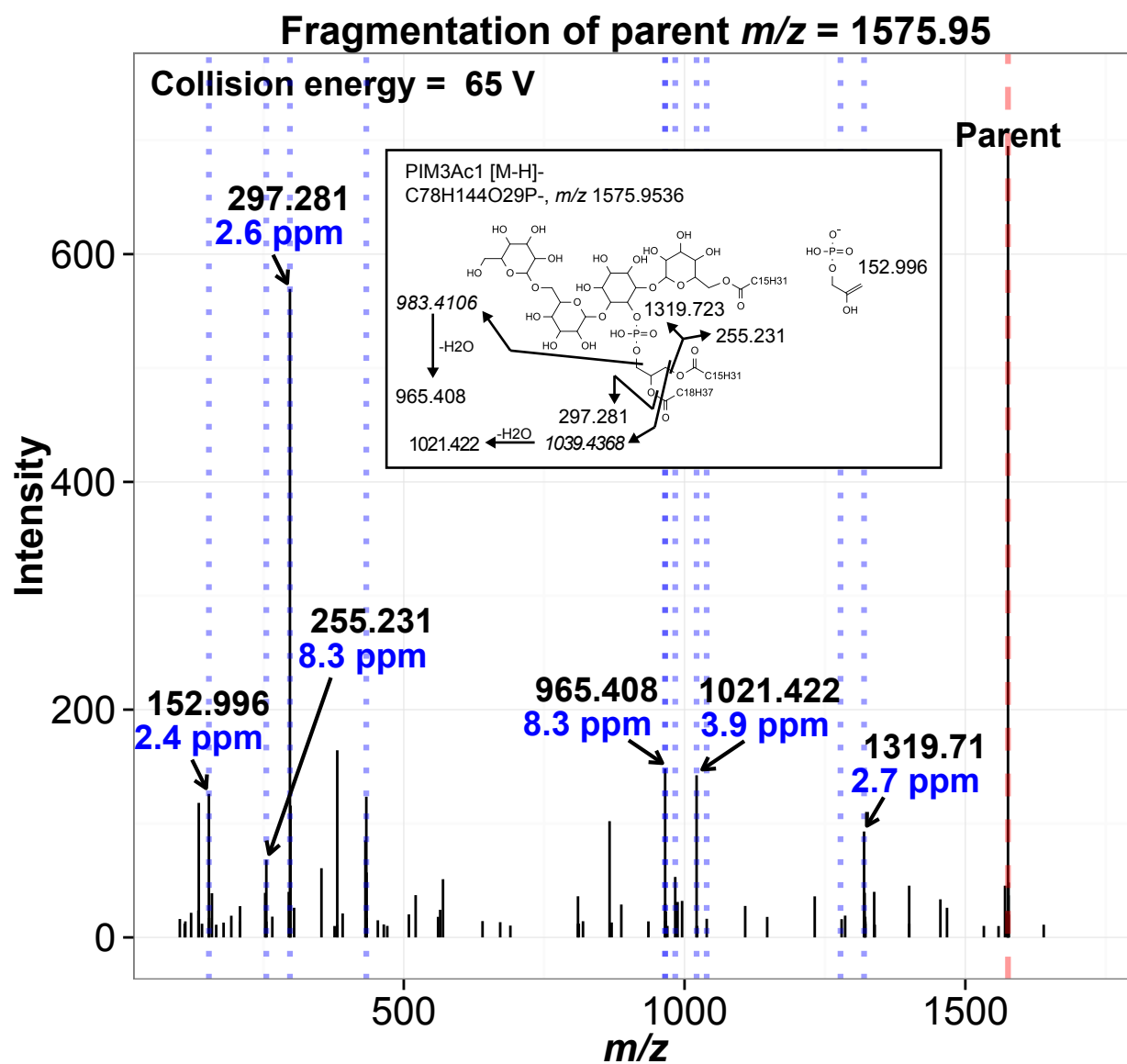


Figure S8

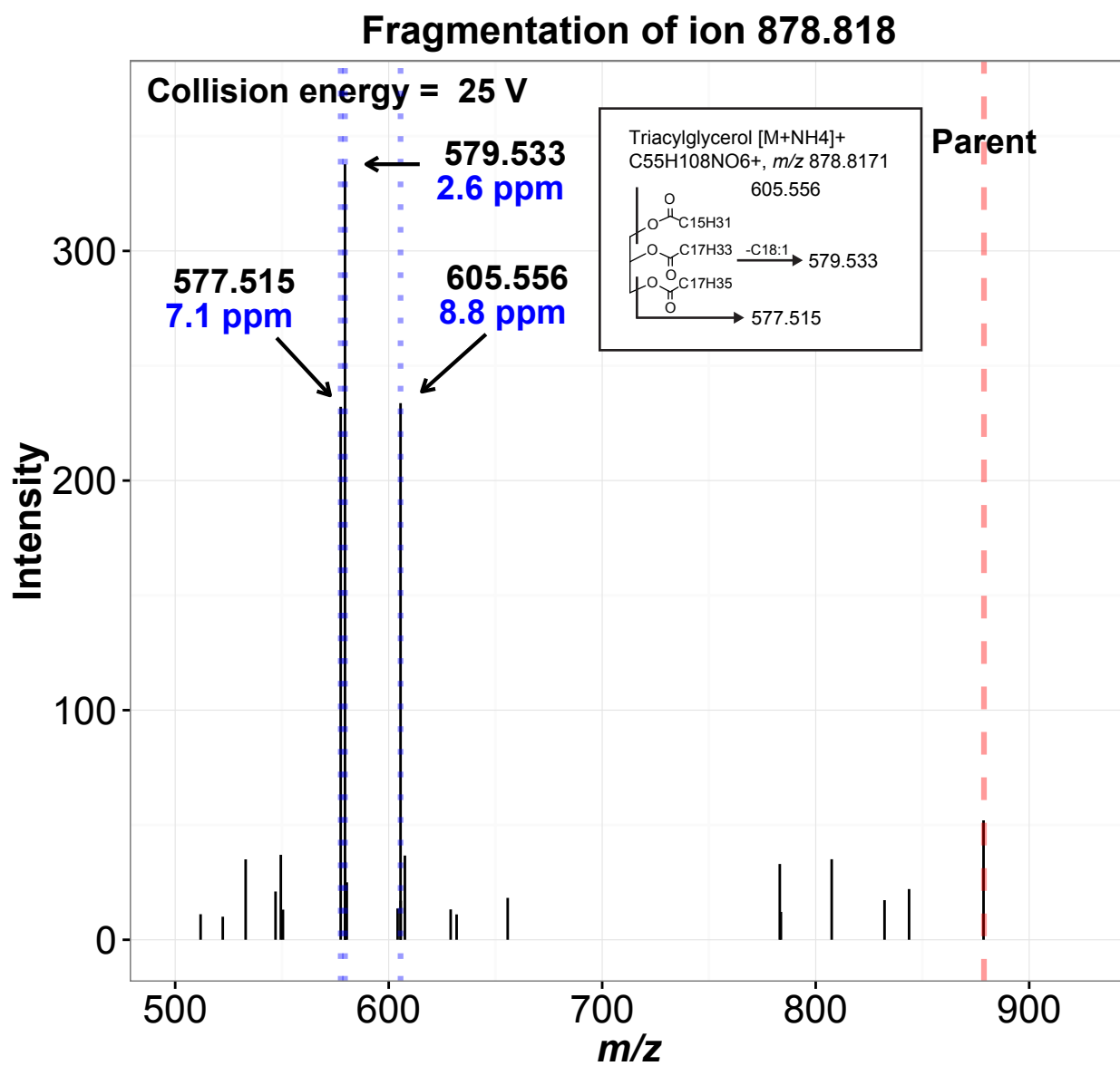


Figure S9

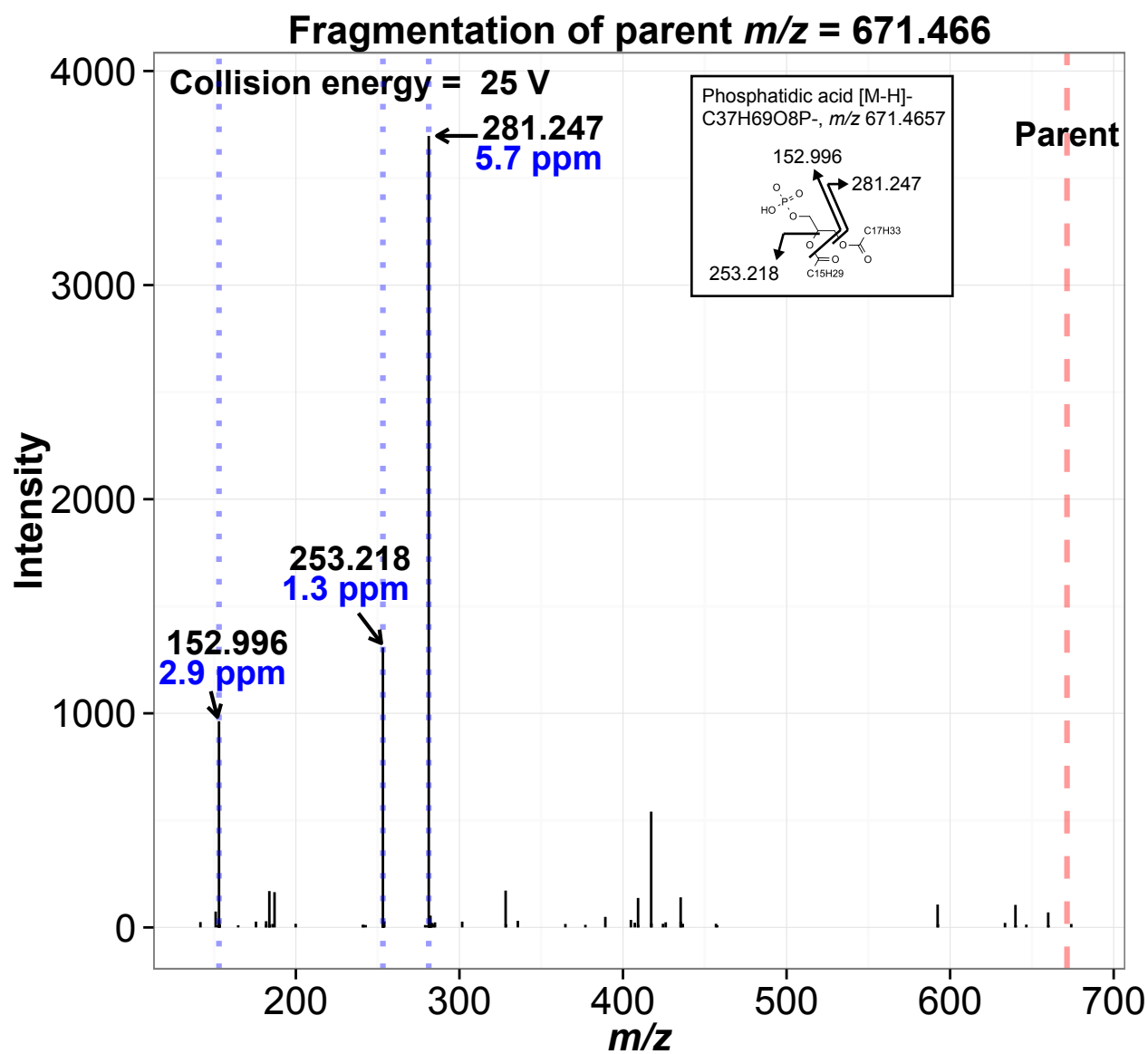


Figure S10

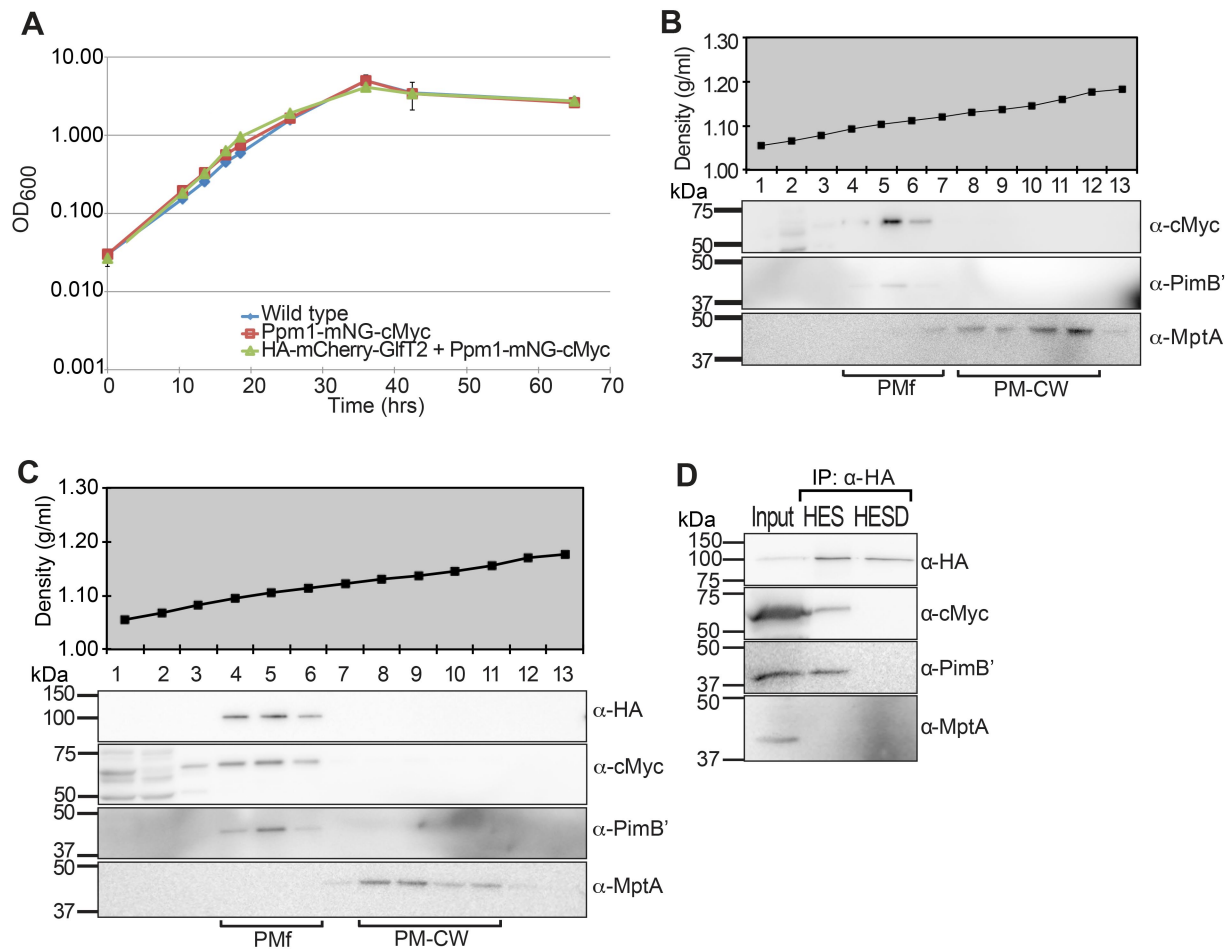


Figure S11

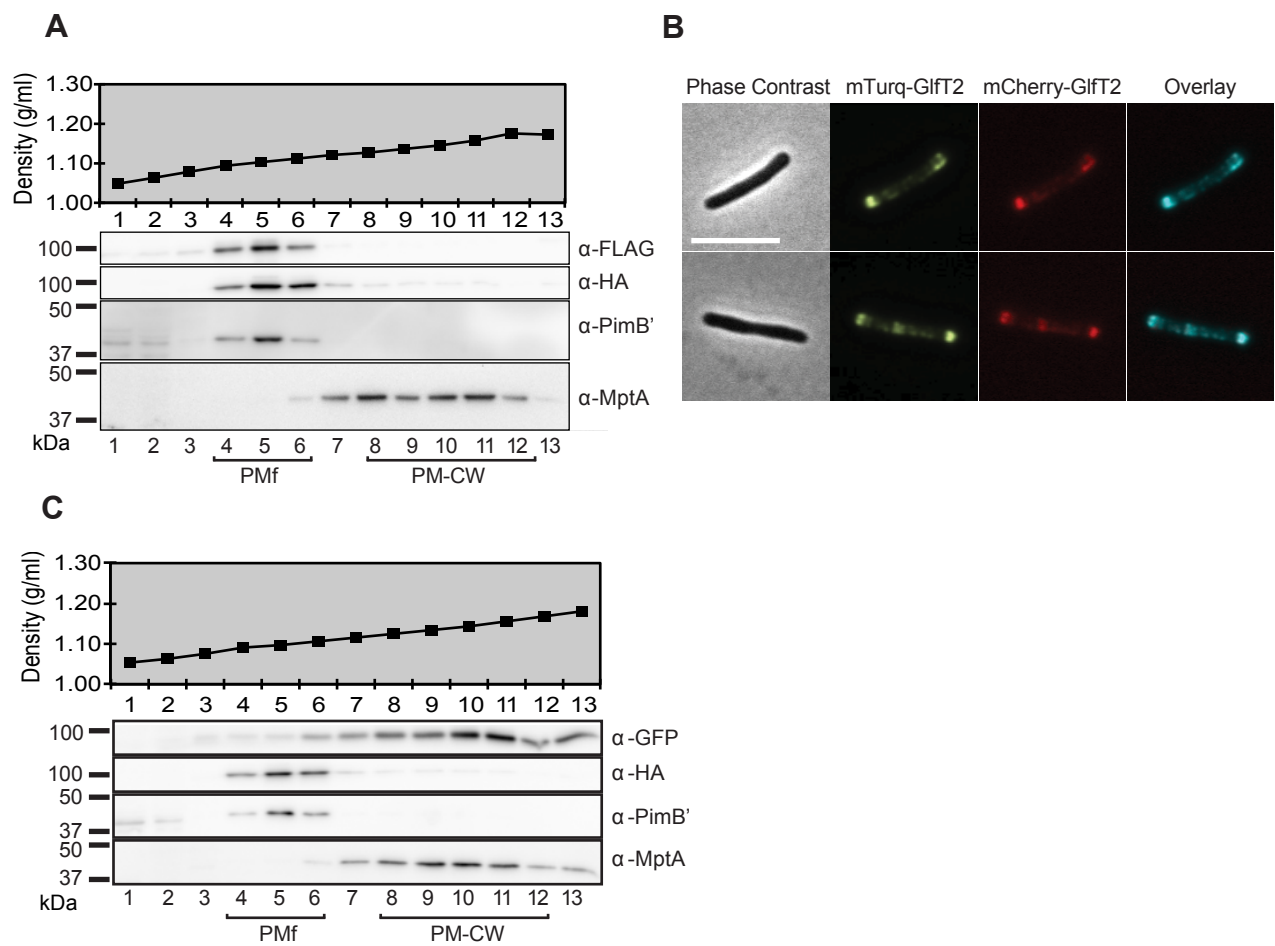


Figure S12

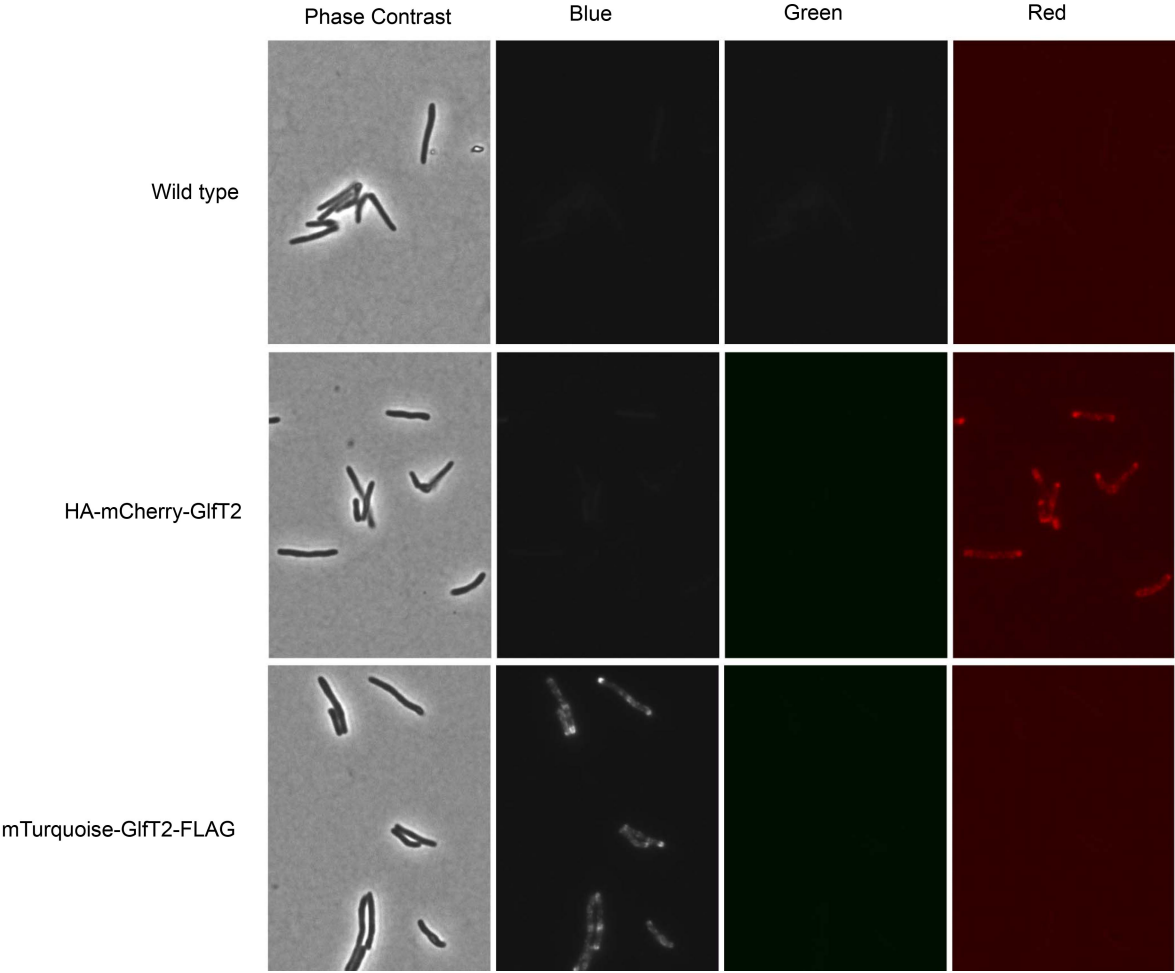


Figure S13

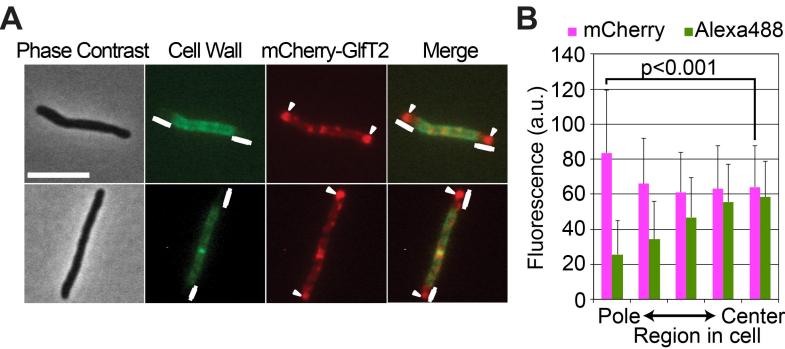
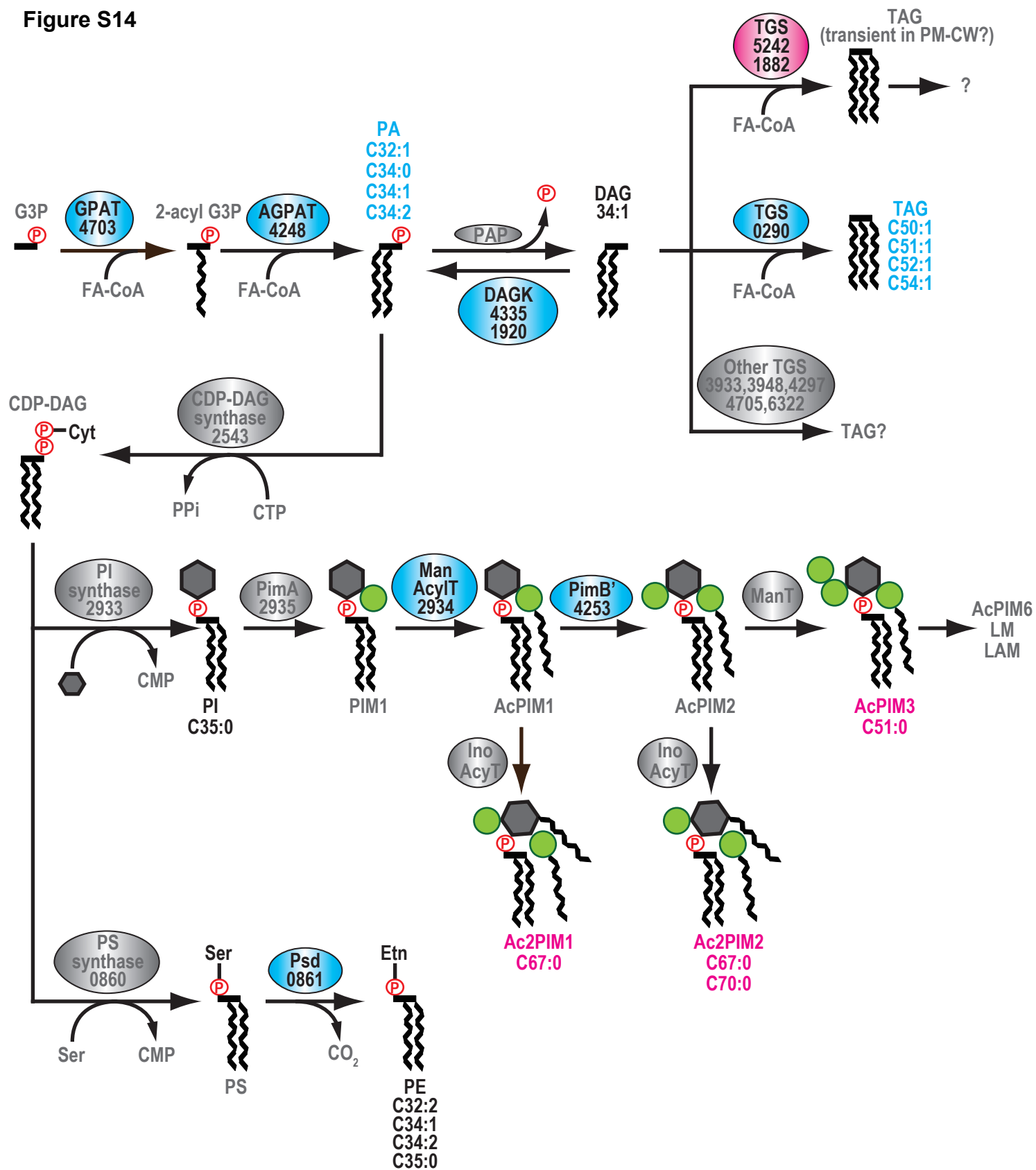


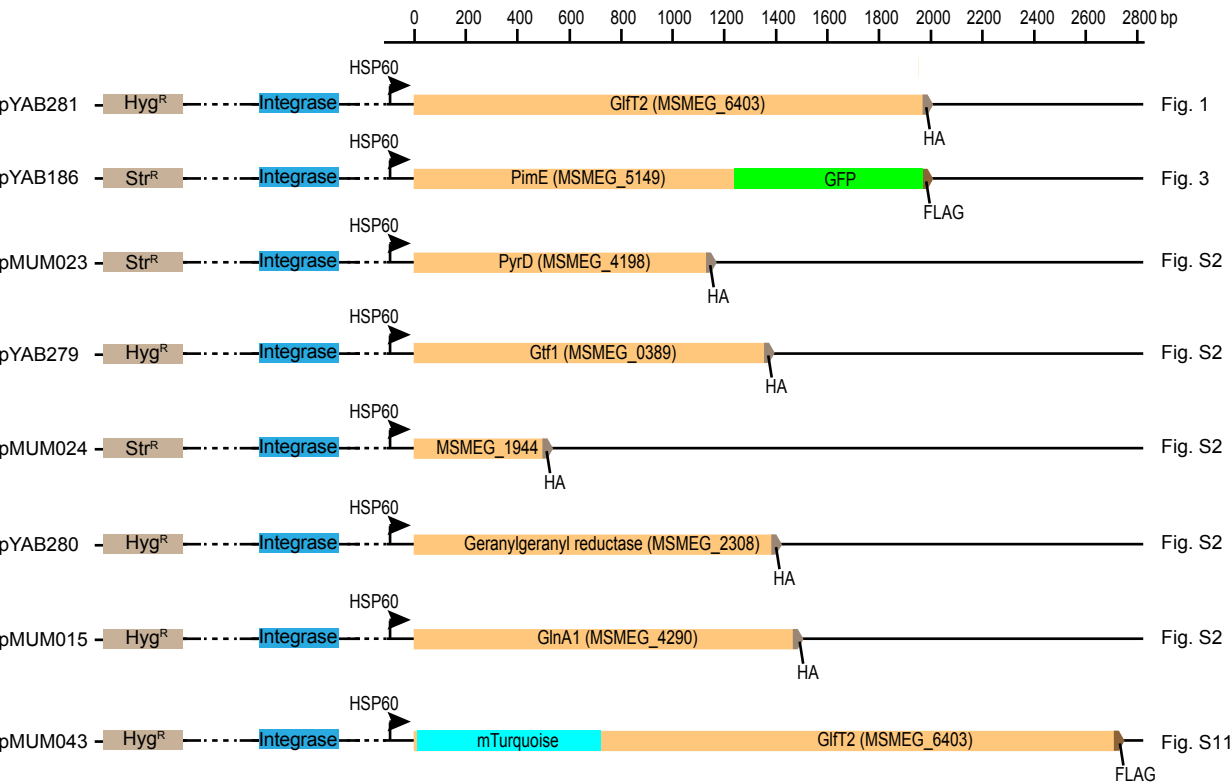
Figure S14



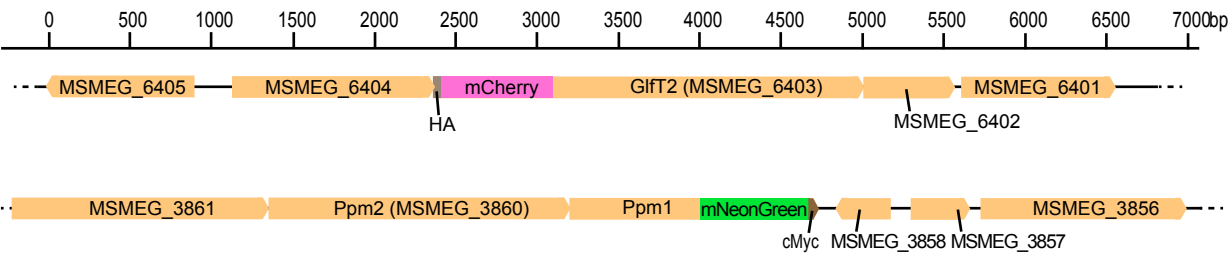
	Cyt Cytidine		Lipids in Cyan	Lipids in PMf lipidome
	Ser Serine		Lipids in Magenta	Lipids in PM-CW lipidome
	Etn Ethanolamine		Lipids in Black	Lipids in both lipidomes
			Lipids in Gray	Lipids not identified

Figure S15

A Expression vectors for mycobacteriophage L5 integration



B Markerless gene replacement at endogenous loci



```
#####
#
# Dataset S3
# Analysis of PMf vs PM-CW at MS1 and MS2 for Yasu Morita
#
# This is an R script containing all of the parameters and statistical
# analysis for the lipidomics experiments.
# Running the script will produce Figure 2A, Figures S5-9, Table S3 and
# Dataset S2.
#
# To repeat the analysis using this script:
# 1) Request the mass spec data files, or start with XCMS diffreport
# files provided
# which contain the peaks picked by XCMS using the parameters found
# found in xcmsFun to run the MS1 analysis
#
# 2) Download the MycoMap and MycoMass databases, available at:
# http://www.brighamandwomens.org/research/depts/medicine/rheumatology/
# Labs/Moody/default.aspx
# The Excel files must be renamed/reformatted to R-friendly files.
# or use the provided, curated files
#
# 3) The tandem MS data are provided as a RDS file.
# Raw data are available on request
#
# 4) Set file paths that make sense for your computer.
# Put the data files in a folder to call.
#
#####
#

# Packages

library(xcms)
library(aroma.light)
library(corrplot)
library(limma)
library(gplots)
library(ggplot2)
library(grid)
library(gridExtra)
library(stringr)
library(extrafont)

#####
#
```

```
# Functions
```

```
# xcmsFun:
```

```
# function to perform xcms peak picking, grouping, etc
```

```
# input a file path and xcms setting
```

```
# output an xcmsSet object (default)
```

```
# or diffreport object (if diffreport=TRUE)
```

```
# the object must be assigned or it will just print
```

```
# The diffreport can be saved instead of made into an object
```

```
# with save.table=TRUE and filebase=some_name
```

```
# saved reports can be read with:
```

```
# pl.xcms <- read.table("~/filebase.tsv")
```

```
# The diffreport MUST have levels to compare, set by level.1 and level.2
```

```
xcmsFun <-
```

```
function(path=".", method='centWave', ppm=10, snthresh=5,  
peakwidth=c(20,120), prefilter=c(1,100), noise=0, integrate=1,  
fitgauss=FALSE, verbose.columns=TRUE, bw=30, minfrac=0.5, mzwid=0.25,  
level.1=1, level.2=2, diffreport=FALSE, save.table=FALSE, filebase=NULL  
)  
{
```

```
  mzdatafiles <- list.files(path, recursive=TRUE, full.names=TRUE)
```

```
  xset <- xcmsSet(mzdatafiles, method=method, ppm=ppm, snthresh=snthresh,  
peakwidth=peakwidth, prefilter=prefilter, noise=noise, integrate=integrate,  
fitgauss=fitgauss, verbose.columns=verbose.columns)
```

```
  xset
```

```
  xset <- group(xset, bw=bw, minfrac=minfrac, mzwid=mzwid)
```

```
  xset
```

```
  xset2 <- retcor(xset, family="s", plotype="m")
```

```
  xset2 <- group(xset2, bw=bw, minfrac=minfrac, mzwid=mzwid)
```

```
  xset3 <- fillPeaks(xset2) # method="chrom"
```

```
  xset3
```

```
  if(diffreport==FALSE) {
```

```
    return(xset3)
```

```
  }
```

```
  else if(save.table==TRUE) {
```

```
    diffreport(xset3, levels(sampclass(xset3))[level.1],  
levels(sampclass(xset3))[level.2], filebase, eicmax=0, eicwidth=0,
```

```

sortpval=TRUE, value="into", metlin=0.003)
}

else {
  diffreport(xset3, levels(sampclass(xset3))[level.1],
levels(sampclass(xset3))[level.2], eicmax=0, eicwidth=0, sortpval=TRUE,
value="into", metlin=0.003)
}
}

```

Replace zeros with local minima

```

imputeZerosUnifMin <-
function(x, output="log2")
{
  xm <- apply(x,2, function(z)
  {
    m <- min(z[z!=0])
    z[z==0]<-runif(sum(z==0),0,m)
    z
  })
  m <- apply(x,2, function(z) min(z[z!=0]))
  if(output == "min") {
    list(xm=xm,m=m)
  }
  else if(output == "log2") {
    log2(xm)
  }
}

```

Normalize:

Default fullQuantileNorm works well

```

runNorm <-
function(x, method = "fullQuantileNorm")
{
  if(method == "fullQuantileNorm") {
    # Full Quantile
    fullQuantileNorm <- function(x, y=rep(1,ncol(x)))
    {
      x.fq <- x
      for(i in unique(y))
        x.fq[,y==i] <- normalizeQuantile(x[,y==i], robust=TRUE)
    }
  }
}

```

```

        x.fq
    }
    fullQuantileNorm(x)
}

else if(method == "globalScalingNorm") {
    #Global
    globalScalingNorm <- function(x, y=rep(1,ncol(x)), f = function(z)
quantile(z, probs=0.75))
    {
        n <- apply(x, 2, f)
        for(i in unique(y))
            n[y==i] <- n[y==i] - f(x[,y==i])
        scale(x,center=n,scale=FALSE)
    }
    globalScalingNorm(x)
}

else if(method == "pc1Norm") {
    # PC1
    pc1Norm <- function(x, y=rep(1,ncol(x)))
    {
        x.pc1 <- x
        for(i in unique(y))
        {
            s <- svd(x[,y==i])
            u <- s$u[,1]
            x.pc1[,y==i] <- apply(x[,y==i], 2, function(z) (z -
sum(z*u)*u/sum(u^2)))
        }
        x.pc1
    }
    pc1Norm(x)
}
}

```

Database matching:

```

dbMatch <-
function(x, mzmedcol, db, dbnames, dbclass, dbcol, mass_tolerance=0.012)
{
    {

        num.peaks <- dim(x)[1]
        mzs <- x[,mzmedcol]
    }
}

```



```

    dbs <- as.data.frame(db[,dbcol])

    matches <- function(i) {
      which((dbs>=(mzs[i]-mass_tolerance))&(dbs<=(mzs[i]
+mass_tolerance))), arr.ind=T)
    }
    m <- lapply(1:num.peaks, matches)
    met.index <- lapply(1:num.peaks, function(i) m[[i]][,1])
    num.matches <- sapply(m, nrow)

    metabolite <- sapply(1:num.peaks, function(i) if(num.matches[i]>=1)
paste(db[(met.index[[i]][1:length(met.index[[i]])])), dbnames], collapse=";
") else paste("ND"))
    class <- sapply(1:num.peaks, function(i) if(num.matches[i]>=1)
paste(unique(db[(met.index[[i]][1:length(met.index[[i]])])), dbclass]),
collapse="; ") else paste(""))
  }
  data.frame(cbind(x, num.matches, metabolite, class))
}

dbMatch_2 <-
function(x, mzmedcol, db, dbnames, dbclass, dbcol, mass_tolerance=0.012,
rtmedcol, x_rt=60, dbrt, rt_tolerance=3)
{
  {
    if(missing(dbrt)) {
      print("No retention time detected. You should be using dbMatch
not dbMatch_2")
      num.peaks <- dim(x)[1]
      mzs <- x[,mzmedcol]
      dbs.mz <- as.data.frame(db[,dbcol])

      matches <- function(i) {
        which((dbs.mz>=(mzs[i]-mass_tolerance))&(dbs.mz<=(mzs[i]
+mass_tolerance))), arr.ind=T)
      }
      m <- lapply(1:num.peaks, matches)
      met.index <- lapply(1:num.peaks, function(i) m[[i]][,1])
      num.matches <- sapply(m, nrow)
    }

    else {
      num.peaks <- dim(x)[1]
      mzs <- x[,mzmedcol]
      dbs.mz <- as.data.frame(db[,dbcol])

```

```

    rts <- x[, (rtmedcol)]/x_rt
    dbs.rt <- as.data.frame(db[, dbrt])

    matches <- function(i) {
      which(
        ((dbs.mz>=(mzs[i]-mass_tolerance))&(dbs.mz<=(mzs[i]
+mass_tolerance)))) &
        ((dbs.rt>=(rts[i]-rt_tolerance))&(dbs.rt<=(rts[i]
+rt_tolerance))))),
      arr.ind=T)
    }

    m <- lapply(1:num.peaks, matches)
    met.index <- lapply(1:num.peaks, function(i) m[[i]][,1])
    num.matches <- sapply(m, nrow)
  }

  metabolite <- sapply(1:num.peaks, function(i) if(num.matches[i]>=1)
paste(db[(met.index[[i]][1:length(met.index[[i]])]), dbnames], collapse=";
") else paste("ND"))
  class <- sapply(1:num.peaks, function(i) if(num.matches[i]>=1)
paste(unique(db[(met.index[[i]][1:length(met.index[[i]])]), dbclass]),
collapse="; ") else paste(""))
  }
  if(x_rt==60) {
    print("dbMatch2 assumes 1) rtmedcol is in seconds, the xcms default &
2) dbrt is in minutes, the MycoMap default. Set x_rt=1 if mzmed is in
minutes.")
  }
  data.frame(cbind(x, num.matches, metabolite, class))
}

```

```

# Get the fragments from a list of scans that meet specified criteria
# produces a list
# assign to an object
# output annotated with the scan number

```

```

# Tandem MS functions:

```

```

# Return the depth of a list
list.depth <- function(this, thisdepth = 0) {

```

```
# http://stackoverflow.com/a/13433689/1270695
if(!is.list(this)) {
  return(thisdepth)
} else {
  return(max(unlist(lapply(this, list.depth, thisdepth = thisdepth+1))))
}
}
```

```
# ppm calculator
```

```
ppm <- function (expected, observed)
  {((observed-expected)/expected)*1000000}
```

```
# Get all scans from an xcmsRaw MSn=TRUE object or list
# produces a list
# assign to an object
# output annotated with the sample, parent ion and collision energy
```

```
peakList <-
function(xr) {
  if(is.list(xr) == FALSE)
  {
    lapply(1:length(xr@msnScanindex), function(i) {
      peaks <- getMsnScan(xr, i);
      ll <- order(peaks[,1]);
      peaks <- peaks[ll,];
      sample <- i
      mass0 <- xr@msnPrecursorMz[i];
      col0 <- xr@msnCollisionEnergy[i];
      scan0 <- xr@msnScanindex[i];
      rt0 <- xr@msnRt[i];

      if(length(ll)==1) {print("Error: getScan < 2")}
      }

      else {cbind(sample, peaks, mass0, col0, scan0, rt0)}
    })
  }

  else if(is.list(xr[[1]]) == FALSE)
  {
    lapply(1:length(xr), function(i) {
```

```

lapply(1:length(xr[[i]]@msnScanindex), function(j) {

    peaks <- getMsnScan(xr[[i]], j);
    ll <- order(peaks[,1]);
    peaks <- peaks[ll,];
    sample <- i
    mass0 <- xr[[i]]@msnPrecursorMz[j];
    col0 <- xr[[i]]@msnCollisionEnergy[j];
    scan0 <- xr[[i]]@msnScanindex[j];
    rt0 <- xr[[i]]@msnRt[j];

    if(length(ll)==1) {print("Error: getScan < 2")}
    }

    else {cbind(sample, peaks, mass0, col0, scan0, rt0)
    }
})
}

else
{
print("Assuming this is parsed MSn data")
lapply(1:length(xr), function(i) {
    lapply(1:length(xr[[i]]), function(j) {
        lapply(1:length(xr[[i]][[j]]@scanindex), function(k) {

            peaks <- getScan(xr[[i]][[j]], k);
            ll <- order(peaks[,1]);
            peaks <- peaks[ll,];
            sample <- i
            # stats <- names(xr[[i]])[j]
            mass0 <- xr[[i]][[j]]@msnPrecursorMz[k];
            col0 <- xr[[i]][[j]]@msnCollisionEnergy[k];
            scan0 <- xr[[i]][[j]]@scanindex[k];
            rt0 <- xr[[i]][[j]]@scantime[k];

            if(length(ll)==1) {print("Error: getScan < 2")}
            }

            else {cbind(sample, peaks, mass0, col0, scan0, rt0)
            }

        })
    })
})
}

```

```

    }
}

```

Recursively remove nulls from a list

```
is.NullOb <- function(x) is.null(x) | all(sapply(x, is.null))
```

Recursively step down into list, removing all such objects

```
rmNullObs <- function(x) {
  x <- Filter(Negate(is.NullOb), x)
  lapply(x, function(x) if (is.list(x)) rmNullObs(x) else x)
}
```

search for fragments using a reference list

```
peakFrgs <-
function(peak.list, tags, tag.width) {

  if(list.depth(peak.list) == 1) {
    lapply(1:length(peak.list), function(j) {
      lapply(1:dim(peak.list[[j]])[1], function(i) {
        if(any(peak.list[[j]][i,2] >= tags - tag.width &
              peak.list[[j]][i,2] <= tags + tag.width) == TRUE) {
          frgs <- append(peak.list[[j]][i,], j, 1)
          names(frgs)[2] <- "scan"
          frgs
        }
      })
    })
  }

  else if(list.depth(peak.list) == 2) {
    lapply(1:length(peak.list), function(k) {
      lapply(1:length(peak.list[[k]]), function(j) {
        lapply(1:dim(peak.list[[k]][[j]])[1], function(i) {
          if(any(peak.list[[k]][[j]][i,2] >= tags - tag.width &
                peak.list[[k]][[j]][i,2] <= tags + tag.width) ==
TRUE) {
            frgs <- append(peak.list[[k]][[j]][i,], j, 1)
            names(frgs)[2] <- "scan"
            frgs
          }
        })
      })
    })
  }
}
```

```

    })
  }
  else {print("peak.list structure not recognized")}
}

peakFragmentsPPM <-
function(peak.list, tags, ppmCut) {

  if(list.depth(peak.list) == 1) {
    lapply(1:length(peak.list), function(j) {
      lapply(1:dim(peak.list)[[j]])[1], function(i) {

        ppm <- sapply(1:length(tags), function(x)
          abs(ppm(tags[x], peak.list[[j]][i,2])))

        ppm <- min(ppm)

        if(ppm <= ppmCut)
          {
            frags <- append(peak.list[[j]][i,], j, 1)
            names(frags)[2] <- "scan"
            frags <- append(frags, ppm, 3)
            names(frags)[4] <- "ppm"
            frags
          }
      })
    })
  }

  else if(list.depth(peak.list) == 2) {
    lapply(1:length(peak.list), function(k) {
      lapply(1:length(peak.list)[[k]], function(j) {
        lapply(1:dim(peak.list)[[k]][[j]])[1], function(i) {

          ppm <- sapply(1:length(tags), function(x)
            abs(ppm(tags[x], peak.list[[k]][[j]][i,2])))

          ppm <- min(ppm)

          if(ppm <= ppmCut)
            {
              frags <- append(peak.list[[k]][[j]][i,], j, 1)
              names(frags)[2] <- "scan"
              frags <- append(frags, ppm, 3)
              names(frags)[4] <- "ppm"
            }
        }
      })
    })
  }
}

```

```

                                frags
                                }
                            })
                        })
                    }
else {print("peak.list structure not recognized")}

}

```

```

# Make fragmentation plots
# Need a table using the peakList format: requires mz, intensity, mass0 &
col0
# Need a vector of reference frgements. Set to the lower limit if not
needed.
# Add ppm calculations for fragments by setting ppmThresh
# set to < 1 if not desired
# label positions can be adjusted with xadj/yadj

fragPlot <- function(data, limits=c(100, 1800), refFrgs, ppmThresh=10,
ppm_xadj=0, ppm_yadj=0, ppm_labadj=-50, cutoff=FALSE, anno_x=limits[1],
anno_y=10, anno_xadj=0, anno_yadj=0, mar=c(1,1,1,1), arrow_x=NULL,
arrow_xend=NULL, arrow_y=NULL, arrow_yend=NULL, ...) {

    ppm <- sapply(1:length(refFrgs), function(i) abs(ppm(refFrgs[i],
data$mz)))
    ppm <- apply(ppm, 1, min)
    mzCut <- data.frame(data, ppm)

    mzCut <- mzCut[mzCut$ppm <= ppmThresh,]
    print(mzCut)

    if(is.null(arrow_x))
        arrow_x <- rep(0, dim(mzCut)[1])
    if(is.null(arrow_xend))
        arrow_xend <- rep(0, dim(mzCut)[1])
    if(is.null(arrow_y))
        arrow_y <- rep(0, dim(mzCut)[1])
    if(is.null(arrow_yend))
        arrow_yend <- rep(0, dim(mzCut)[1])

    ggp <- ggplot(data, aes(x=mz, y=intensity, ymin=0, ymax=intensity))
+
    geom_linerange() +

```

```

scale_x_continuous(limits=limits)+
theme_bw() +

geom_vline(xintercept=data$mass0[1], color="red", alpha=0.4,
size=1.1, linetype="dashed") +

annotate("text", x=data$mass0[1], y=max(data$intensity)+10,
label="Parent", size=5, fontface="bold") +

annotate("text", x=anno_x, y=max(data$intensity + 0.05*data
$intensity) + anno_y , label=paste("\nCollision energy = ", data$col0[1],
"V"), size=5, fontface="bold", hjust=0.075, vjust=0) +

theme(plot.title=element_text(size=16, face="bold"),
axis.text=element_text(size=16), axis.title.y=element_text(size=16,
face="bold"), axis.title.x=element_text(size=16, face="bold.italic")) +
xlab("m/z") +
ylab("Intensity") +
ggtitle(paste("Fragmentation of ion", data$mass0[1], sep=" ")) +

geom_vline(xintercept=refFrag, color="blue", alpha=0.4, size=1.1,
linetype="dotted") +

theme(plot.margin = unit(mar, "cm"))

if(is.numeric(ppmThresh) == TRUE) {
  ggp <- ggp + annotate("segment",
x=mzCut$mz + 35 + arrow_x,
xend=mzCut$mz + 10 + arrow_xend,
y=mzCut$intensity + arrow_y,
yend=mzCut$intensity + arrow_yend,
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.75)
}

if(is.numeric(ppmThresh) == TRUE) {
  ggp <- ggp +
  annotate("text",
x=mzCut$mz + ppm_xadj + 20,
y=mzCut$intensity + ppm_yadj + 20,
label=format.AsIs(round(mzCut$mz, 3), nsmall=3), size=5,
fontface="bold", color="black") +
  annotate("text",
x=mzCut$mz + ppm_xadj + 20,
y=mzCut$intensity + ppm_yadj + ppm_labadj,
label=paste(format.AsIs(round(mzCut$ppm, 1), nsmall=1), "ppm ",

```



```
sep=" "), size=5, fontface="bold", color="blue")
}
```

```
if(is.numeric(cutoff) == TRUE) {
  ggp <- ggp +
    annotate("text", x=data$mz[data$intensity > cutoff] +
anno_xadj, y=data$intensity[data$intensity > cutoff] + anno_yadj + 5,
label=
      format.AsIs(round(data$mz[data$intensity > cutoff], 3),
nsmall=3), size=5, fontface="bold")
}
```

```
ggp
```

```
}
```

```
#####
#
```

```
# Reference Data
```

```
# Expected tandem MS fragments from MycoMap
```

```
MyMapFrag <-
```

```
list(
```

```
  PE.n = list(ID="PE",
    mode="Negative",
    formula="C40H79N08P-",
    parentMass=732.55488,
    notes=NA,
    fragments=c("C16H3102-"=255.2330, "C19H3702-"=297.2799,
"C24H49N07P-"=494.3252)
  ),
```

```
  PIM2Ac2.n = list(ID="PIM2Ac2",
    mode="Negative",
    formula="C91H170025P-",
    parentMass=1694.17743,
    notes=c("2C16", "2C19"),
    fragments=c("C3H605P-"=152.99583, "C16H3102-"=255.23295,
"C19H3702-"=297.27990, "C22H4206P-"=433.27245,
"C50"=803.34719+(1694.17743-1413.90082), # extrapolated from
PIM2AC1
"C50"=821.35775+(1694.17743-1413.90082), # extrapolated from
PIM2AC1
```

```

      "C53"=859.37340+(1694.17743-1413.90082), # extrapolated from
PIM2AC1
      "C53"=877.38344+(1694.17743-1413.90082), # extrapolated from
PIM2AC1
      "C53H170020P-"=1083.62381, "C72H132023P-"=1395.89025,
"C75H138023P-"=1437.93720)
    ),

```

```

PIM3Ac1.n = list(ID="PIM3Ac1",
  mode="Negative",
  formula="C78H145029P-",
  parentMass=1575.9536,
  notes="Extrapolated from other PIMs; 2C16, 2C19",
  fragments=c("C3H605P-"=152.99583, "C16H3102-"=255.23295,
"C19H3702-"=297.27990, "C22H4206P-"=433.27245,
  "C40H60024P-"=803.34719+(1575.9536-1413.90082),
  "C40H62025P-"=821.35775+(1575.9536-1413.90082),
  "C43H64025P-"=859.37340+(1575.9536-1413.90082),
  "C43H66026P-"=877.38344+(1575.9536-1413.90082),
  "C43H170024P-"=1083.62381+(1575.9536-1694.17743),
  "C53H96022P-"=1115.61363+(1575.9536-1413.90082),
  "C56H102022P-"=1157.66059+(1575.9536-1413.90082))
),

```

```

PA_671.n = list(ID="PI",
  mode="Negative",
  formula="C37H6908P-",
  parentMass=671.46573,
  notes=NA,
  fragments=c("C3H605P-"=152.9958, "C16H2902-"=253.2173,
"C18H3302-"=281.2486)
),

```

```

PA_673.n = list(ID="PI",
  mode="Negative",
  formula="C37H7108P-",
  parentMass=673.48138,
  notes=NA,
  fragments=c("C3H605P-"=152.9958, "C16H2902-"=253.2173,
"C16H3102-"=255.23295, "C18H3302-"=281.2486, "C18H3502-"=283.2643)
)
)

```

```

#####
#

```

```

# Read in MS1 data

# If starting from raw MS data
# peakpicking of negative and positive MS1 data is done by:

# xset_n <- xcmsFun("~/093015JM/Negative")
# xset_n
# xset_p <- xcmsFun("~/093015JM/Positive")
# xset_p

# Save grouped peaks
# pl.xcms_n <- diffreport(xset_n, levels(sampclass(xset_n))[1],
  levels(sampclass(xset_n))[2], eicmax=0, eicwidth=0, sortpval=TRUE,
  value="into", metlin=0.003, filebase="093015_pl_n")

# If starting from the provided diffreport files
# Once the above steps are complete, reload the data with:
# Read in grouped peaks from file
# Read in to convert factors to numeric
pl.xcms_n <- read.table("~/Documents/Projects/Morita_lipidomics/
  093015_pl_n.tsv", colClasses=c(rep("character", 15), rep("numeric", 16)))
paste(dim(pl.xcms_n)[1], "ions were found in the negative mode")

# Note:
# If reading data from Dataset S2, the

# Repeat for positive mode
# pl.xcms_p <- diffreport(xset_p, levels(sampclass(xset_p))[1],
  levels(sampclass(xset_p))[2], eicmax=0, eicwidth=0, sortpval=TRUE,
  value="into", metlin=0.003, filebase="093015_pl_p")
pl.xcms_p <- read.table("~/Documents/Projects/Morita_lipidomics/
  093015_pl_p.tsv", colClasses=c(rep("character", 15), rep("numeric", 16)))
paste(dim(pl.xcms_p)[1], "ions were found in the positive mode")

# Rename for +/- and reformat numbers
pl.xcms_n[,1] <- paste(pl.xcms_n[,1], "n", sep="")
pl.xcms_p[,1] <- paste(pl.xcms_p[,1], "p", sep="")

pl.xcms_n[,5] <- as.numeric(pl.xcms_n[,5])
pl.xcms_n[,8] <- as.numeric(pl.xcms_n[,8])
pl.xcms_p[,5] <- as.numeric(pl.xcms_p[,5])
pl.xcms_p[,8] <- as.numeric(pl.xcms_p[,8])

```

```

# Impute and normalize
set.seed(818)
pl.il_n <- imputeZerosUnifMin(pl.xcms_n[,15:30])
rownames(pl.il_n) <- pl.xcms_n[,1]
set.seed(3937)
pl.il_p <- imputeZerosUnifMin(pl.xcms_p[,15:30])
rownames(pl.il_p) <- pl.xcms_p[,1]

#####
#

# Quality control monitoring:

# The impounded measurements can be flagged and counted.
metimp_n <- t(sapply(1:dim(pl.xcms_n)[1], function(i) pl.xcms_n[i,
  15:30]==0))
numimp_n <- sapply(1:dim(pl.xcms_n)[1], function(i)
  length(which(metimp_n[i,]=="TRUE"))))
lcids_n <- sapply(1:dim(pl.xcms_n)[1], function(i)
  paste(as.character(colnames(pl.xcms_n[, 15:30])[which(pl.xcms_n[i,
  15:30]==0)]), collapse="; "))

metimp_p <- t(sapply(1:dim(pl.xcms_p)[1], function(i) pl.xcms_p[i,
  15:30]==0))
numimp_p <- sapply(1:dim(pl.xcms_p)[1], function(i)
  length(which(metimp_p[i,]=="TRUE"))))
lcids_p <- sapply(1:dim(pl.xcms_p)[1], function(i)
  paste(as.character(colnames(pl.xcms_p[, 15:30])[which(pl.xcms_p[i,
  15:30]==0)]), collapse="; "))

# Generate a new table with information about impounded measurements
xcmsImp_n <- data.frame(name=pl.xcms_n$name, number_imp=numimp_n,
  impounded=lcids_n)
xcmsImp_p <- data.frame(name=pl.xcms_p$name, number_imp=numimp_p,
  impounded=lcids_p)

# One sample is impounded 3595 times:
# summary(xcmsImp_p$lcids) # X093015JM57_YM_cw4_p.mzdata
# summary(xcmsImp_n$lcids) # X093015JM07_YM_f4_n.mzdata is impounded 2620
# Nothing else is close
# both are BioRep 4's

```

```
# Biological replicate 4 is of lower quality, but equally so for PMf and PM-  
CW
```

```
# Removing it might improve variance, but is not advisable
```

```
# Negative and positive must be kept separate through this step
```

```
pl.fq_n <- runNorm(pl.il_n)
```

```
pl.fq_p <- runNorm(pl.il_p)
```

```
# Correlations
```

```
pl_n.cor <- cor(pl.fq_n)
```

```
pl_p.cor <- cor(pl.fq_p)
```

```
  # add color ramps
```

```
  cor.ramp1 <- colorpanel(3, "yellow", "orange", "red")
```

```
  cor.ramp2 <- colorpanel(2, "white", "yellow")
```

```
  cor.ramp3 <- colorpanel(9, "black", "grey", "white")
```

```
  cor.col <- c(cor.ramp3, cor.ramp2, cor.ramp1)
```

```
corrplot(pl_p.cor, cl.lim=c(min(pl_p.cor),1), insig="pch", pch=4,  
pch.cex=0.6, type="upper", col=cor.col, method="ellipse", tl.cex=1,  
tl.srt=45, title="Correlation between samples after normalization",  
mar=c(1,0,1,0))
```

```
# Need to rename and reorder columns in +/- to match
```

```
# Reorder both by sample number and replicate
```

```
pl.fq_n <- pl.fq_n[,c(1,4,3,2,7,6,5,8,10,12,11,9,15,16,14,13)]
```

```
pl.fq_p <- pl.fq_p[,c(1,2,4,3,7,8,5,6,9:12,15,16,14,13)]
```

```
# Pull full file names
```

```
files_n <- colnames(pl.fq_n)
```

```
files_p <- colnames(pl.fq_p)
```

```
files_n <- gsub(".*J", "J", files_n)
```

```
files_p <- gsub(".*J", "J", files_p)
```

```
files_n <- gsub("_n.mzdata", "", files_n)
```

```
files_p <- gsub("_p.mzdata", "", files_p)
```

```
files_n <- gsub("_YM_", "", files_n)
```

```
files_p <- gsub("_YM_", "", files_p)
```

```
fileIds <- paste(files_n, files_p, sep=".")
```

```
pl.fq <- rbind(pl.fq_n, pl.fq_p)
```

```
colnames(pl.fq) <- fileIds
```

```
# Correlations
```

```
pl.cor <- cor(pl.fq)
```

```
colnames(pl.cor) <- c("cw1a", "cw2a", "cw3a", "cw4a", "cw1b", "cw2b",  
"cw3b", "cw4b", "f1a", "f2a", "f3a", "f4a", "f1b", "f2b", "f3b",  
"f4b")
```

```
rownames(pl.cor) <- c("cw1a", "cw2a", "cw3a", "cw4a", "cw1b", "cw2b",  
"cw3b", "cw4b", "f1a", "f2a", "f3a", "f4a", "f1b", "f2b", "f3b",  
"f4b")
```

```
pl.ord <- c("cw1a", "cw1b", "cw2a", "cw2b", "cw3a", "cw3b", "cw4a", "cw4b",  
"f1a", "f1b", "f2a", "f2b", "f3a", "f3b", "f4a", "f4b")
```

```
# add color ramps
```

```
cor.ramp1 <- colorpanel(3, "yellow", "orange", "red")
```

```
cor.ramp2 <- colorpanel(2, "white", "yellow")
```

```
cor.ramp3 <- colorpanel(9, "black", "grey", "white")
```

```
cor.col <- c(cor.ramp3, cor.ramp2, cor.ramp1)
```

```
corrplot(pl.cor[pl.ord, pl.ord], cl.lim=c(min(pl.cor),1), insig="pch",  
pch=4, pch.cex=0.6, type="upper", col=cor.col, method="ellipse", tl.cex=1,  
tl.srt=45, title="Correlation between samples after normalization",  
mar=c(1,0,1,0))
```

```
# Truncate bottom lipids
```

```
# pl.mn <- apply(pl.fq, 1, mean)
```

```
# pl.hi <- pl.fq[pl.mn >= 15,]
```

```
# pl.corHi <- cor(pl.hi)
```

```
# corrplot(pl.corHi[pl.ordHi, pl.ordHi], cl.lim=c(min(pl.corHi),1),  
insig="pch", pch=4, pch.cex=0.6, type="upper", col=cor.col,  
method="ellipse", tl.cex=1, tl.srt=45, title="Correlation between samples  
after normalization", mar=c(1,0,1,0))
```

```
# Not driven by low level lipids
```

```
# use avearrays to average arrays
```

```
pl.fqa <- pl.fq
```

```
colnames(pl.fqa) <- c("cw1", "cw2", "cw3", "cw4", "cw1", "cw2", "cw3",  
"cw4", "f1", "f2", "f3", "f4", "f1", "f2", "f3", "f4")
```

```
pl.fqa <- avearrays(pl.fqa)
```

```

pl_a.cor <- cor(pl.fqa)
pl_a.ord <- c("cw1", "cw2", "cw3", "cw4", "f1", "f2", "f3", "f4")

# add color ramps
cor.ramp1 <- colorpanel(3, "yellow", "orange", "red")
cor.ramp2 <- colorpanel(4, "white", "yellow")
cor.ramp3 <- colorpanel(8, "black", "grey", "white")
cor.col <- c(cor.ramp3, cor.ramp2, cor.ramp1)

corrplot(pl_a.cor[pl_a.ord, pl_a.ord], cl.lim=c(min(pl_a.cor),1),
  insig="pch", pch=4, pch.cex=0.6, type="lower", col=cor.col,
  method="ellipse", tl.cex=1, tl.srt=45, title="Correlation between samples
  after normalization", mar=c(1,0,1,0))

# Biorep 4 still correlates well and should be included

#####
#

# Statistical analysis

# rename and split names into design matrix
targets.a <- data.frame(name=colnames(pl.fqa))
targets.a[,2] <- substr(targets.a[,1], nchar(as.character(targets.a[,
1]))-2, nchar(as.character(targets.a[,1]))-1)
targets.a[,3] <- substr(targets.a[,1], nchar(as.character(targets.a[,1])),
  nchar(as.character(targets.a[,1]))+1)
names(targets.a) <- c("Name", "Fraction", "BioRep")

fraction <- as.numeric(factor(targets.a$Fraction))
biorep <- as.numeric(factor(targets.a$BioRep))

# Now build a model matrix

design <- model.matrix(~ fraction + biorep)
# design

# Use limma to build, fit and test a model
fit <- lmFit(pl.fqa, design)
fit <- eBayes(fit)

```

```

tT_pl <- topTable(fit, coef="fraction", number=dim(fit$t)[1])

results <- decideTests(fit)
sum <- summary(results)

# QC: a QQ-plot
par(mar=c(7,7,7,7))
qqt(fit$t[,2], main="QQ-Plot, Paired Sample X Fraction")
abline(0,1)
# Looks good.

# Build a fully annotated object
# Stats in tT_pl; intensities in pl.fq & pl.fqa; mz and rt in pl.xcms;
# information about impounded measurements is in xcmsImp_n
# Add putative ids from MycoMap and MycoMass

# Add impounded flags to xcms summary stats
pl.ss_n <- merge(pl.xcms_n[,1:13], xcmsImp_n, by.x=1, by.y=1)
pl.ss_p <- merge(pl.xcms_p[,1:13], xcmsImp_p, by.x=1, by.y=1)

# An xcms summary object
xcmsStats <- rbind(pl.ss_n, pl.ss_p)

# Add xcms to tT object
tT_xcms <- merge(tT_pl, xcmsStats, by.x=0, by.y=1)

# Add peak intensities to the topTable
tT_pia <- merge(tT_xcms, pl.fqa, by.x=1, by.y=0)
tT_pi <- merge(tT_pia, pl.fq, by.x=1, by.y=0)

# Output full table of information
# This is Dataset S2
write.csv(tT_pi, "~/Lipidomics_data_w_stats.csv")

#####
#

# Database matching:

```



```

# A note about database matching:
# Matches between ions and the m/z and/or retention time of a known
  compound
# are putative at best.
# Database matching is performed below only to choose targets
# that are subsequently validated by tandem MS.

# MycoMap and MycoMass are available at:
# http://www.brighamandwomens.org/research/depts/medicine/rheumatology/
  Labs/Moody/default.aspx
# Note: the downloaded versions must be curated to R readable versions
# by changing the names and saving as .csv, etc
# ie names(MycoMap)
# > names(MycoMap)
# [1] "Class"          "Subclass"       "Formula"
# [4] "Detected.m_z"   "Ion_mode"       "Calculated_mz_pos"
# [7] "Calculated_mz_neg" "Dm_amu"         "RT_min"
# [10] "RT_min2"        "RT_dev"

# Match to mass and retention time
# MycoMap matching
MycoMap <- read.csv("~/Documents/Projects/Morita_lipidomics/MycoMap_R.csv",
  header=TRUE)

tT_ID.n <- dbMatch_2(tT_pi[substr(tT_pi[,1], nchar(as.character(tT_pi[,
  1])), nchar(as.character(tT_pi[,1]))+1) == "n",], mzmedcol=11, db=MycoMap,
  dbnames=1, dbclass=3, dbcol=7, rtmedcol=14, dbrt=9, rt_tolerance=8)
tT_ID.p <- dbMatch_2(tT_pi[substr(tT_pi[,1], nchar(as.character(tT_pi[,
  1])), nchar(as.character(tT_pi[,1]))+1) == "p",], mzmedcol=11, db=MycoMap,
  dbnames=1, dbclass=3, dbcol=6, rtmedcol=14, dbrt=9, rt_tolerance=8)

tT_ID <- rbind(tT_ID.n, tT_ID.p)

# Annotate by MycoMass
MycoMass <- read.csv("~/Documents/Projects/Moody/MycoMass_R.csv",
  header=TRUE)
names(MycoMass)[1:8] <- c("Abbreviation", "Mass", "[M+H]+", "[M+NH4]+", "[M
+Na]+", "[M-H]-", "[M+HCOO]-", "[M+CH3COO]-")

tT_An <- dbMatch(tT_ID[substr(tT_ID[,1], nchar(as.character(tT_ID[,1])),
  nchar(as.character(tT_ID[,1]))+1) == "n",], 11, MycoMass, 9, 13, 6:8)
tT_Ap <- dbMatch(tT_ID[substr(tT_ID[,1], nchar(as.character(tT_ID[,1])),
  nchar(as.character(tT_ID[,1]))+1) == "p",], 11, MycoMass, 9, 13, 3:5)

```

```

tT_An <- rbind(tT_An, tT_Ap)

# Who has MycoMap matches
tT_wID <- tT_ID[(tT_ID$num.matches != 0),]
tT_wID[,c(1:2,11,47)]

# Who has MycoMass matches
tT_wAn <- tT_An[(tT_An$num.matches.1 != 0),]
tT_wAn[,c(1:3,6,11,50:51)]
tT_wAn[,c(1,50)]

# Add intensity, fc, qc cutoffs
# tT_lfc: - is PM-CW, + is PMf
tT_cut <- tT_wAn[tT_wAn$logFC >= 1 & tT_wAn$AveExpr >= 15 & tT_wAn
$number_imp <= 8,c(1:3,6,11,50)]

#####
#

# Figure 2A

# Volcano plot
# color flag set to 2-fold change
cf <- tT_pl[abs(tT_pl$logFC) >=1 & tT_pl$adj.P.Val <=0.05, ]

made.cut <- cbind(PMf.up=sum(tT_pl$logFC >=1 & tT_pl$adj.P.Val <=0.05),
PMCW.up=sum(tT_pl$logFC <=-1 & tT_pl$adj.P.Val <=0.05),
ns=sum(abs(tT_pl$logFC) < 1 | tT_pl$adj.P.Val > 0.05),
total=dim(tT_pl)[1]
)

Row.names <- row.names(tT_pl)

# Map points
PE <- tT_pl[row.names(tT_pl)=="M733T1200n", ]
PIM3Ac1 <- tT_pl[row.names(tT_pl)=="M1576T1991n", ] # C51:0
PA_671 <- tT_pl[row.names(tT_pl)=="M671T454n", ]
PA_673 <- tT_pl[row.names(tT_pl)=="M673T428n", ]

# More map points
PA_645 <- tT_pl[row.names(tT_pl)=="M645T457n", ]
PA_675 <- tT_pl[row.names(tT_pl)=="M675T419n", ]

TAG_850 <- tT_pl[row.names(tT_pl)=="M851T183p", ]

```

```

TAG_865 <- tT_pl[row.names(tT_pl)=="M865T360p", ]
TAG_879 <- tT_pl[row.names(tT_pl)=="M879T185p", ]
TAG_906 <- tT_pl[row.names(tT_pl)=="M907T184p", ]

PI_851 <- tT_pl[row.names(tT_pl)=="M852T1484n", ]

PE_715 <- tT_pl[row.names(tT_pl)=="M715T1159n", ]
PE_686 <- tT_pl[row.names(tT_pl)=="M686T1174n", ]
PE_716 <- tT_pl[row.names(tT_pl)=="M717T1192n", ]

DAG_612 <- tT_pl[row.names(tT_pl)=="M613T185p", ]

Ac2PIM_1492 <- tT_pl[row.names(tT_pl)=="M1492T1504p", ] # C67:0
Ac2PIM2_1671 <- tT_pl[row.names(tT_pl)=="M1671T1501p", ] # C67:0
Ac2PIM2_1713 <- tT_pl[row.names(tT_pl)=="M1713T1469p", ] # C70:0 pos mode
1694

# font_import()
# loadfonts()

# pdf("~/Documents/Projects/Moody/Yasu/YM_Figure2a.pdf", family="ArialMT",
height=6.83, width=6.83)

ggplot(tT_pl, aes(x=logFC, y=-log10(adj.P.Val), color=AveExpr)) +
  geom_point(shape=1) +
  scale_color_gradient(low="grey", high="black", name="Intensity\n
(log2)") +
  theme(legend.key = element_rect(color="white")) +
  geom_hline(yintercept=1.30103, linetype="dotted") +
  geom_vline(xintercept=1, linetype="dotted") +
  geom_vline(xintercept=-1, linetype="dotted") +
  theme_bw() +
  theme(panel.border=element_rect(color="black"),
panel.grid.major=element_blank(), panel.grid.minor=element_blank()) +
  theme(legend.position=c(0,0), legend.justification=c(0.05,0.05),
legend.background=element_rect(fill=(alpha=0))) +

  theme(plot.margin = unit(c(1,1,1,1), "cm")) +
  theme(axis.title=element_text(size=16, face="bold"),
axis.text=element_text(size=16), axis.title.x=element_text(hjust=0.5),
axis.title.y=element_text(hjust=0.5), legend.text=element_text(size=16),
legend.title=element_text(size=16), plot.title=element_text(size=16,
hjust=0.25, face="bold")) +

  xlab(paste(" Fold change (log2)")) +

```

```

ylab("Significance (-log10 p value)") +
ggtitle(paste(made.cut[2], "up in PM-CW", "n =", made.cut[4],
", made.cut[1], "up in PMf", sep=" ")) +

geom_point(data=PE, aes(x=PE$logFC, yend=-log10(PE$adj.P.Val), color=PE
$AveExpr), shape=16) +
annotate("segment", x=PE$logFC-1.8, xend=PE$logFC-0.3, y=-log10(PE
$adj.P.Val), yend=-log10(PE$adj.P.Val), arrow=arrow(length=unit(0.2,
"cm")), color="white", size=1) +
annotate("segment", x=PE$logFC-1.8, xend=PE$logFC-0.3, y=-log10(PE
$adj.P.Val), yend=-log10(PE$adj.P.Val), arrow=arrow(length=unit(0.2,
"cm")), color="black", size=0.7) +
annotate("text", x=PE$logFC, y=-log10(PE$adj.P.Val)-0.035,
label=paste("PE C35:0 (B)"), size=5.5, hjust=1.35, fontface="bold") +

geom_point(data=PIM3Ac1, aes(x=PIM3Ac1$logFC, yend=-
log10(PIM3Ac1$adj.P.Val), color=PIM3Ac1$AveExpr), shape=16) +
annotate("segment", x=PIM3Ac1$logFC-1.8, xend=PIM3Ac1$logFC-0.3, y=-
log10(PIM3Ac1$adj.P.Val), yend=-log10(PIM3Ac1$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
annotate("segment", x=PIM3Ac1$logFC-1.8, xend=PIM3Ac1$logFC-0.3, y=-
log10(PIM3Ac1$adj.P.Val), yend=-log10(PIM3Ac1$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
annotate("text", x=PIM3Ac1$logFC, y=-log10(PIM3Ac1$adj.P.Val),
label=paste("AcPIM3 C51:0 (D)"), size=5.5, hjust=1.25, fontface="bold") +

geom_point(data=PA_671, aes(x=PA_671$logFC, yend=-
log10(PA_671$adj.P.Val), color=PA_671$AveExpr), shape=16) +
annotate("segment", x=PA_671$logFC+1.8, xend=PA_671$logFC+0.3, y=-
log10(PA_671$adj.P.Val), yend=-log10(PA_671$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
annotate("segment", x=PA_671$logFC+1.8, xend=PA_671$logFC+0.3, y=-
log10(PA_671$adj.P.Val), yend=-log10(PA_671$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
annotate("text", x=PA_671$logFC, y=-log10(PA_671$adj.P.Val),
label=paste("PA C34:2 (F)"), size=5.5, hjust=-0.35, fontface="bold") +

geom_point(data=PA_673, aes(x=PA_673$logFC, yend=-
log10(PA_673$adj.P.Val), color=PA_673$AveExpr), shape=16) +
annotate("segment", x=PA_673$logFC+1.8, xend=PA_673$logFC+0.3, y=-
log10(PA_673$adj.P.Val)-0.04, yend=-log10(PA_673$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
annotate("segment", x=PA_673$logFC+1.8, xend=PA_673$logFC+0.3, y=-
log10(PA_673$adj.P.Val)-0.04, yend=-log10(PA_673$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
annotate("text", x=PA_673$logFC, y=-log10(PA_673$adj.P.Val)-0.04,

```

```
label=paste("PA C34:1"), size=5.5, hjust=-0.5, fontface="bold") +
```

```
  geom_point(data=PA_645, aes(x=PA_645$logFC, yend=-  
log10(PA_645$adj.P.Val), color=PA_645$AveExpr), shape=16) +  
  annotate("segment", x=PA_645$logFC+1.8, xend=PA_645$logFC+0.3, y=-  
log10(PA_645$adj.P.Val)+0.04, yend=-log10(PA_645$adj.P.Val),  
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +  
  annotate("segment", x=PA_645$logFC+1.8, xend=PA_645$logFC+0.3, y=-  
log10(PA_645$adj.P.Val)+0.04, yend=-log10(PA_645$adj.P.Val),  
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +  
  annotate("text", x=PA_645$logFC, y=-log10(PA_645$adj.P.Val)+0.04,  
label=paste("PA C32:1"), size=5.5, hjust=-0.5, fontface="bold") +
```

```
  geom_point(data=PA_675, aes(x=PA_675$logFC, yend=-  
log10(PA_675$adj.P.Val), color=PA_675$AveExpr), shape=16) +  
  annotate("segment", x=PA_675$logFC+1.8, xend=PA_675$logFC+0.3, y=-  
log10(PA_675$adj.P.Val)+0.04, yend=-log10(PA_675$adj.P.Val)+0.01,  
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +  
  annotate("segment", x=PA_675$logFC+1.8, xend=PA_675$logFC+0.3, y=-  
log10(PA_675$adj.P.Val)+0.04, yend=-log10(PA_675$adj.P.Val)+0.01,  
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +  
  annotate("text", x=PA_675$logFC, y=-log10(PA_675$adj.P.Val)+0.04,  
label=paste("PA C34:0"), size=5.5, hjust=-0.5, fontface="bold") +
```

```
  geom_point(data=TAG_850, aes(x=TAG_850$logFC, yend=-  
log10(TAG_850$adj.P.Val), color=TAG_850$AveExpr), shape=16) +  
  annotate("segment", x=TAG_850$logFC+1.8, xend=TAG_850$logFC+0.3, y=-  
log10(TAG_850$adj.P.Val)+0.05, yend=-log10(TAG_850$adj.P.Val)+0.01,  
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +  
  annotate("segment", x=TAG_850$logFC+1.8, xend=TAG_850$logFC+0.3, y=-  
log10(TAG_850$adj.P.Val)+0.05, yend=-log10(TAG_850$adj.P.Val)+0.01,  
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +  
  annotate("text", x=TAG_850$logFC, y=-log10(TAG_850$adj.P.Val)+0.05,  
label=paste("TAG C50:1"), size=5.5, hjust=-0.4, fontface="bold") +
```

```
  geom_point(data=TAG_865, aes(x=TAG_865$logFC, yend=-  
log10(TAG_865$adj.P.Val), color=TAG_865$AveExpr), shape=16) +  
  annotate("segment", x=TAG_865$logFC+1.8, xend=TAG_865$logFC+0.3, y=-  
log10(TAG_865$adj.P.Val)-0.03, yend=-log10(TAG_865$adj.P.Val),  
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +  
  annotate("segment", x=TAG_865$logFC+1.8, xend=TAG_865$logFC+0.3, y=-  
log10(TAG_865$adj.P.Val)-0.03, yend=-log10(TAG_865$adj.P.Val),  
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +  
  annotate("text", x=TAG_865$logFC, y=-log10(TAG_865$adj.P.Val)-0.03,  
label=paste("TAG C51:1"), size=5.5, hjust=-0.45, fontface="bold") +
```

```

geom_point(data=TAG_879, aes(x=TAG_879$logFC, yend=-
log10(TAG_879$adj.P.Val), color=TAG_879$AveExpr), shape=16) +
  annotate("segment", x=TAG_879$logFC+1.8, xend=TAG_879$logFC+0.3, y=-
log10(TAG_879$adj.P.Val)-0.07, yend=-log10(TAG_879$adj.P.Val)-0.01,
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
  annotate("segment", x=TAG_879$logFC+1.8, xend=TAG_879$logFC+0.3, y=-
log10(TAG_879$adj.P.Val)-0.07, yend=-log10(TAG_879$adj.P.Val)-0.01,
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
  annotate("text", x=TAG_879$logFC, y=-log10(TAG_879$adj.P.Val)-0.07,
label=paste("TAG C52:1 (E)"), size=5.5, hjust=-0.3, fontface="bold") +

```

```

geom_point(data=TAG_906, aes(x=TAG_906$logFC, yend=-
log10(TAG_906$adj.P.Val), color=TAG_906$AveExpr), shape=16) +
  annotate("segment", x=TAG_906$logFC+1.8, xend=TAG_906$logFC+0.3, y=-
log10(TAG_906$adj.P.Val), yend=-log10(TAG_906$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
  annotate("segment", x=TAG_906$logFC+1.8, xend=TAG_906$logFC+0.3, y=-
log10(TAG_906$adj.P.Val), yend=-log10(TAG_906$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
  annotate("text", x=TAG_906$logFC, y=-log10(TAG_906$adj.P.Val),
label=paste("TAG C54:1"), size=5.5, hjust=-0.4, fontface="bold") +

```

```

geom_point(data=PI_851, aes(x=PI_851$logFC, yend=-
log10(PI_851$adj.P.Val), color=PI_851$AveExpr), shape=16) +
  annotate("segment", x=PI_851$logFC+1.8, xend=PI_851$logFC+0.3, y=-
log10(PI_851$adj.P.Val), yend=-log10(PI_851$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
  annotate("segment", x=PI_851$logFC+1.8, xend=PI_851$logFC+0.3, y=-
log10(PI_851$adj.P.Val), yend=-log10(PI_851$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
  annotate("text", x=PI_851$logFC, y=-log10(PI_851$adj.P.Val),
label=paste("PI C35:0 (C)"), size=5.5, hjust=-0.4, fontface="bold") +

```

```

geom_point(data=PE_715, aes(x=PE_715$logFC, yend=-
log10(PE_715$adj.P.Val), color=PE_715$AveExpr), shape=16) +
  annotate("segment", x=PE_715$logFC-1.8, xend=PE_715$logFC-0.3, y=-
log10(PE_715$adj.P.Val)+0.03, yend=-log10(PE_715$adj.P.Val)+0.01,
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
  annotate("segment", x=PE_715$logFC-1.8, xend=PE_715$logFC-0.3, y=-
log10(PE_715$adj.P.Val)+0.03, yend=-log10(PE_715$adj.P.Val)+0.01,
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
  annotate("text", x=PE_715$logFC, y=-log10(PE_715$adj.P.Val)+0.03,
label=paste("PE C34:2"), size=5.5, hjust=1.5, fontface="bold") +

```

```

geom_point(data=PE_686, aes(x=PE_686$logFC, yend=-
log10(PE_686$adj.P.Val), color=PE_686$AveExpr), shape=16) +

```



```

    annotate("segment", x=PE_686$logFC-1.8, xend=PE_686$logFC-0.3, y=-
log10(PE_686$adj.P.Val)-0.13, yend=-log10(PE_686$adj.P.Val)-0.03,
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
    annotate("segment", x=PE_686$logFC-1.8, xend=PE_686$logFC-0.3, y=-
log10(PE_686$adj.P.Val)-0.13, yend=-log10(PE_686$adj.P.Val)-0.03,
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
    annotate("text", x=PE_686$logFC, y=-log10(PE_686$adj.P.Val)-0.13,
label=paste("PE C32:2"), size=5.5, hjust=1.5, fontface="bold") +

```

```

    geom_point(data=PE_716, aes(x=PE_716$logFC, yend=-
log10(PE_716$adj.P.Val), color=PE_716$AveExpr), shape=16) +
    annotate("segment", x=PE_716$logFC+1.8, xend=PE_716$logFC+0.3, y=-
log10(PE_716$adj.P.Val)-0.14, yend=-log10(PE_716$adj.P.Val)-0.03,
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
    annotate("segment", x=PE_716$logFC+1.8, xend=PE_716$logFC+0.3, y=-
log10(PE_716$adj.P.Val)-0.14, yend=-log10(PE_716$adj.P.Val)-0.03,
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
    annotate("text", x=PE_716$logFC, y=-log10(PE_716$adj.P.Val)-0.14,
label=paste("PE C34:1"), size=5.5, hjust=-0.5, fontface="bold") +

```

```

    geom_point(data=DAG_612, aes(x=DAG_612$logFC, yend=-
log10(DAG_612$adj.P.Val), color=DAG_612$AveExpr), shape=16) +
    annotate("segment", x=DAG_612$logFC+1.8, xend=DAG_612$logFC+0.3, y=-
log10(DAG_612$adj.P.Val), yend=-log10(DAG_612$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="white", size=1) +
    annotate("segment", x=DAG_612$logFC+1.8, xend=DAG_612$logFC+0.3, y=-
log10(DAG_612$adj.P.Val), yend=-log10(DAG_612$adj.P.Val),
arrow=arrow(length=unit(0.2, "cm")), color="black", size=0.7) +
    annotate("text", x=DAG_612$logFC, y=-log10(DAG_612$adj.P.Val),
label=paste("DAG C34:1"), size=5.5, hjust=-0.5, fontface="bold") +

```

```

    geom_point(data=Ac2PIM_1492, aes(x=Ac2PIM_1492$logFC, yend=-
log10(Ac2PIM_1492$adj.P.Val), color=Ac2PIM_1492$AveExpr), shape=16) +
    annotate("segment", x=Ac2PIM_1492$logFC-1.8,
xend=Ac2PIM_1492$logFC-0.3, y=-log10(Ac2PIM_1492$adj.P.Val)+0.11, yend=-
log10(Ac2PIM_1492$adj.P.Val)+0.03, arrow=arrow(length=unit(0.2, "cm")),
color="white", size=1) +
    annotate("segment", x=Ac2PIM_1492$logFC-1.8,
xend=Ac2PIM_1492$logFC-0.3, y=-log10(Ac2PIM_1492$adj.P.Val)+0.11, yend=-
log10(Ac2PIM_1492$adj.P.Val)+0.03, arrow=arrow(length=unit(0.2, "cm")),
color="black", size=0.7) +
    annotate("text", x=Ac2PIM_1492$logFC, y=-log10(Ac2PIM_1492$adj.P.Val)
+0.11, label=paste("Ac2PIM C67:0"), size=5.5, hjust=1.35, fontface="bold")
+

```

```

    geom_point(data=Ac2PIM2_1671, aes(x=Ac2PIM2_1671$logFC, yend=-

```

```

log10(Ac2PIM2_1671$adj.P.Val), color=Ac2PIM2_1671$AveExpr), shape=16) +
  annotate("segment", x=Ac2PIM2_1671$logFC-1.8,
xend=Ac2PIM2_1671$logFC-0.3, y=-log10(Ac2PIM2_1671$adj.P.Val)-0.1, yend=-
log10(Ac2PIM2_1671$adj.P.Val)-0.03, arrow=arrow(length=unit(0.2, "cm")),
color="white", size=1) +
  annotate("segment", x=Ac2PIM2_1671$logFC-1.8,
xend=Ac2PIM2_1671$logFC-0.3, y=-log10(Ac2PIM2_1671$adj.P.Val)-0.1, yend=-
log10(Ac2PIM2_1671$adj.P.Val)-0.03, arrow=arrow(length=unit(0.2, "cm")),
color="black", size=0.7) +
  annotate("text", x=Ac2PIM2_1671$logFC, y=-
log10(Ac2PIM2_1671$adj.P.Val)-0.1, label=paste("Ac2PIM2 C67:0"), size=5.5,
hjust=1.3, fontface="bold") +

```

```

  geom_point(data=Ac2PIM2_1713, aes(x=Ac2PIM2_1713$logFC, yend=-
log10(Ac2PIM2_1713$adj.P.Val), color=Ac2PIM2_1713$AveExpr), shape=16) +
  annotate("segment", x=Ac2PIM2_1713$logFC-1.8,
xend=Ac2PIM2_1713$logFC-0.3, y=-log10(Ac2PIM2_1713$adj.P.Val), yend=-
log10(Ac2PIM2_1713$adj.P.Val), arrow=arrow(length=unit(0.2, "cm")),
color="white", size=1) +
  annotate("segment", x=Ac2PIM2_1713$logFC-1.8,
xend=Ac2PIM2_1713$logFC-0.3, y=-log10(Ac2PIM2_1713$adj.P.Val), yend=-
log10(Ac2PIM2_1713$adj.P.Val), arrow=arrow(length=unit(0.2, "cm")),
color="black", size=0.7) +
  annotate("text", x=Ac2PIM2_1713$logFC, y=-
log10(Ac2PIM2_1713$adj.P.Val), label=paste("Ac2PIM2 C70:0"), size=5.5,
hjust=1.3, fontface="bold")

```

```
# dev.off()
```

```
#####
#
```

```
# Table S3
```

```
# Summary table for keystone lipids
```

```
# Add more info from MycoMap
```

```
tT_wFullID <- merge(tT_wID, MycoMap, by.x=48, by.y=3)
```

```
# Filter classes w/o CID and info not needed
```

```
tT_wFullID <- tT_wFullID[,c(2:4,7,12,1,49:50, 52:54)]
```

```
tT_wFullID <- tT_wFullID[grepl("glyce",tT_wFullID[,7]) |
  grepl("phos",tT_wFullID[,7]),]
```



```

tT_wFullID <- tT_wFullID[!grepl("lyso",tT_wFullID[,8]),]

expected_mz <- sapply(1:length(tT_wFullID[,10]), function(i) {
  if(is.na(tT_wFullID[i,10])==FALSE) {tT_wFullID[i,10]}
  else {tT_wFullID[i,11]}
})

tT_wFullID[,10] <- expected_mz
tT_wFullID <- tT_wFullID[,-11]

names(tT_wFullID)[6] <- "Formula"
names(tT_wFullID)[10] <- "Calculated_mz"


# Read in hits from volcano
volHits <- rbind(DAG_612, PA_645, PA_671, PA_673, PA_675, PE_686, PE_715,
  PE_716, PE, PI_851, Ac2PIM_1492, Ac2PIM2_1671, Ac2PIM2_1713, PIM3Ac1,
  TAG_850, TAG_865, TAG_879, TAG_906)

# Remove MycoMap matches
sT_woMycoMap <- unique(tT_An[tT_An[,1] %in% row.names(volHits)[!
  row.names(volHits) %in% tT_wFullID[,1]],c(1:3, 6, 11, 50:51)])

names(sT_woMycoMap)[6:7] <- c("Class", "Formula")

sT_wMycoMass <- merge(sT_woMycoMap, MycoMass, all.x=TRUE, sort=FALSE) #
  None need a subclass
# but do need the calculated_mz

sT_wMycoMass_mz <- c(sT_wMycoMass[1:5,13], sT_wMycoMass[6,10],
  sT_wMycoMass[7:8,11])

sT_wMycoMass_ion <- c(rep(names(sT_wMycoMass[13]),5),
  names(sT_wMycoMass[10]), rep(names(sT_wMycoMass[11]),2))

sT_woMycoMap$Calculated_mz <- sT_wMycoMass_mz
sT_woMycoMap$Ion_mode <- sT_wMycoMass_ion


# Merge hits with CID and those with MycoMap matches only
sumTab <- merge(tT_wFullID, sT_woMycoMap, all.x=TRUE, all.y=TRUE)

sumTab$CID <- sumTab$Row.names %in% row.names(volHits)

```

```

# fix, shorten names
sumTab$Class <- gsub("phosphatidyl- ethanolamine",
  "phosphatidylethanolamine", sumTab$Class)
sumTab$Class <- gsub("phosphatidyl-ethanolamine",
  "phosphatidylethanolamine", sumTab$Class)
sumTab$Class <- gsub("phosphatidyl- inositol", "phosphatidylinositol",
  sumTab$Class)
sumTab$Class <- gsub("phosphatidyl- inositol", "phosphatidylinositol",
  sumTab$Class)
sumTab$Class <- gsub("phosphatidy- inositol", "phosphatidylinositol",
  sumTab$Class)
sumTab$Class <- gsub("phosphatidy- inositol", "phosphatidylinositol",
  sumTab$Class)
sumTab$Class <- gsub("phosphatidylethanolamines",
  "phosphatidylethanolamine", sumTab$Class)
sumTab$Class <- gsub("phosphatidyl-myo-inositol-dimannosides",
  "phosphatidylinositol dimannoside", sumTab$Class)
sumTab$Class <- gsub("phosphatidyl-myo-inositol-monomannosides",
  "phosphatidylinositol monomannosides", sumTab$Class)
sumTab$Class <- gsub("phosphatidyl-myo-inositol-monomannosides",
  "phosphatidylinositol monomannosides", sumTab$Class)

sumTab$Ion_mode <- gsub("pos/", "", sumTab$Ion_mode)
sumTab$Ion_mode <- gsub("neg/", "", sumTab$Ion_mode)
sumTab$CID <- gsub("FALSE", "N", sumTab$CID)
sumTab$CID <- gsub("TRUE", "Y", sumTab$CID)

sumTab$ppm <- round(ppm(sumTab$Calculated_mz, sumTab$mzmed),2)

sumTab$mzmed <- round(sumTab$mzmed,3)
sumTab$Calculated_mz <- round(sumTab$Calculated_mz,3)
sumTab$logFC <- round(sumTab$logFC,2)
sumTab$AveExpr <- round(sumTab$AveExpr,2)
sumTab$adj.P.Val <- round(sumTab$adj.P.Val,4)

# Reorder and rename

sumTab <- sumTab[,c(7,10,6,1,9,5,12,8,11,2,4,3)]

names(sumTab) <- c("Class", "Subclass", "Formula", "xcms Name", "Expected
m/z", "Observed m/z", "Deviation (ppm)", "Ion", "CID", "Fold change

```

```

(log2)", "Adjusted P value", "Average normalized intensity")

sumTab <- sumTab[order(sumTab$Class),]

# Fix subclass
sumTab$Subclass[is.na(sumTab$Subclass) & grepl("inositol", sumTab$Class)]
<- "diacyl"
sumTab$Subclass[is.na(sumTab$Subclass)] <- ""

# Filter redundant rows
# TAGs with 300 s retention times are too late and 100 s times should be
used
# positive mode PIMs with CID should be used over negative mode w/o

sumTab <- sumTab[!sumTab$"xcms Name" %in% c("M852T1622n", "M851T366p",
"M865T360p", "M879T364p", "M907T361p", "M963T367p", "M1652T1485n",
"M1694T1462n"),]

sumTab

# write.xlsx2(sumTab, "Documents/Projects/Moody/Yasu/TableS3.xlsx",
sheetName="Summary_Table", row.names=FALSE)

#####
#

# Tandem MS analysis

# If starting from raw MSMS data
# mzdatafiles <- list.files("~/Documents/Projects/Morita_lipidomics/
Morita_MSMS", recursive=TRUE, full.names=TRUE)

# xcms
# Raw
# xr <- sapply(1:length(mzdatafiles), function(i) xcmsRaw(mzdatafiles[i],
includeMSn=TRUE))

# saveRDS(xr, "Morita_XCMSRaw")

# If starting from the provided RDS file
xr <- readRDS("~/Documents/Projects/Morita_lipidomics/Morita_XCMSRaw")

```

```

# Get parents and scans
# Get all Msn scans
peak.list <- peakList(xr)

# To match measured fragments to calculated fragments, use a database
# of expected fragments based on MycoMap.
# read in MyMapFragments
MyMapFragments <- readRDS("MyMapFragments")

names(MyMapFragments)

#####
# PE

peak.list.pe <- peakList(xr[[4]])

peak.fragments.pe <- peakFragmentsPPM(peak.list.pe, tags=MyMapFragments$PE.n$fragments,
ppmCut=10)

# Clean the list by removing null scans
found.fragments.pe <- rmNullObs(peak.fragments.pe)
# Split the nested list into a single data frame
found.fragments.pe <- unlist(found.fragments.pe, recursive=FALSE)
all.fragments.pe <- t(sapply(1:length(found.fragments.pe), function(i)
  found.fragments.pe[[i]]))

# scans meeting fragmentation criteria, ie same parent, collision energy,
etc
setList.pe <- all.fragments.pe[(all.fragments.pe[,6] >= 732.4 &
  all.fragments.pe[,6] <= 732.7) # same parent mass
  & all.fragments.pe[,7] == 35 # same collision energy
  ,]
setList.pe

# Plots

refFragments.pe <- c(as.numeric(MyMapFragments$PE.n$parent), as.numeric(MyMapFragments
$PE.n$fragments))

# pdf("~/YM_FigureS5.pdf", family="ArialMT", height=6.83, width=6.83)

```

```
fragPlot(as.data.frame(peak.list.pe[[828]]), limits=c(100, 740),
refFrgs=refFrgs.pe[-1], ppmThresh=10, ppm_xadj=60, ppm_yadj=c(-100, 0,
200), ppm_labadj=-200, mar=c(1,1,1,0), arrow_x=c(0,0,-5), arrow_y=c(-100,
0, 200))
```

```
# dev.off()
```

```
#####
```

```
# Use the alkane series to find more PEs, PI
```

```
PEhead.alkane <- as.numeric(MyMapFrgs$PA_series.n
$fragments[-1]+197.045309488)
```

```
refFrgs.pe_series <- c(140.01182, 196.038, MyMapFrgs$PA_series.n
$fragments[-1], PEhead.alkane)
```

```
# Look for all PEs
```

```
peak.list.pe2 <- lapply(5:7, function(i) peakList(xr[[i]]))
```

```
PEhead.alkane <- as.numeric(MyMapFrgs$PA_series.n
$fragments[-1]+197.045309488)
```

```
refFrgs.pe_series <- c(140.01182, 196.038, MyMapFrgs$PA_series.n
$fragments[-1], PEhead.alkane)
```

```
peak.frgs.pes <- peakFrgsPPM(peak.list.pe2, tags=refFrgs.pe_series,
ppmCut=15)
```

```
found.frgs.pes <- rmNullObs(peak.frgs.pes)
```

```
# Split the nested list into a single data frame
```

```
found.frgs.pes <- unlist(found.frgs.pes, recursive=FALSE)
```

```
found.frgs.pes <- unlist(found.frgs.pes, recursive=FALSE)
```

```
all.frgs.pes <- t(sapply(1:length(found.frgs.pes), function(i)
found.frgs.pes[[i]]))
```

```
f.scan <- paste(all.frgs.pes[,1], all.frgs.pes[,2], sep="_")
```

```
mulFrgs.pes <- all.frgs.pes[duplicated(f.scan) | duplicated(f.scan,
fromLast=TRUE),]
```

```
setList.pes <- mulFrgs.pes[mulFrgs.pes[,9] >= 1000 &
```

```

mulFrag.pes[,9] <= 2000,]

setList.pesID <- dbMatch(setList.pes, 6, MycoMass, 1, 2, 6:8,
  mass_tolerance=0.02)
setList.pesID[grepl("PE", setList.pesID[,11]),]

# Lots of PE 732

# Good enough 686
fragPlot(as.data.frame(peak.list.pe2[[1]][[850]]), limits=c(100, 775),
  refFrag=refFrag.pe_series, ppmThresh=30, mar=c(1,1,1,0))
# support
fragPlot(as.data.frame(peak.list.pe2[[1]][[851]]), limits=c(100, 775),
  refFrag=refFrag.pe_series, ppmThresh=30, mar=c(1,1,1,0))

# Good enough. PE 714
fragPlot(as.data.frame(peak.list.pe2[[2]][[954]]), limits=c(100, 775),
  refFrag=refFrag.pe_series, ppmThresh=30, mar=c(1,1,1,0))

# Good! Add PE 716
fragPlot(as.data.frame(peak.list.pe2[[3]][[1284]]), limits=c(100, 775),
  refFrag=refFrag.pe_series, ppmThresh=30, mar=c(1,1,1,0))
# support
fragPlot(as.data.frame(peak.list.pe2[[3]][[1285]]), limits=c(100, 775),
  refFrag=refFrag.pe_series, ppmThresh=30, mar=c(1,1,1,0))

#####
# PIs

peak.list.pi <- peakList(xr[[6]])

peak.frag.pi <- peakFragPPM(peak.list.pi, tags=MyMapFrag$PI.n$fragments,
  ppmCut=10)

found.frag.pi <- rmNullObs(peak.frag.pi)
# Split the nested list into a single data frame
found.frag.pi <- unlist(found.frag.pi, recursive=FALSE)
all.frag.pi <- t(sapply(1:length(found.frag.pi), function(i)
  found.frag.pi[[i]]))

f.scan <- paste(all.frag.pi[,1], all.frag.pi[,2], sep="_")
mulFrag.pi <- all.frag.pi[duplicated(f.scan) | duplicated(f.scan,
  fromLast=TRUE),]

```

```

setList.pi <- mulFrag.pi[(mulFrag.pi[,9] >= 1200 &
  mulFrag.pi[,9] <= 2000),]
setList.pi

refFrag.pi <- as.numeric(MyMapFrag$PI.n$fragments[-1])

# pdf("~/Documents/Projects/Moody/Yasu/YM_FigureS2C_v2_raw.pdf",
  family="ArialMT", height=6.83, width=6.83)

fragPlot(as.data.frame(peak.list.pi[[936]]), limits=c(140, 900),
  refFrag=refFrag.pi, ppmThresh=10, ppm_labadj=-40,
  ppm_xadj=c(-90,-70, 70, 70, 70),
  ppm_yadj=c(150, 165, 0, 0, 0),
  arrow_x=c(-80,-60, 0, 0, 0),
  arrow_xend=c(-13, -13, 0, 0, 0),
  arrow_y=c(80,90, 0, 0, 0),
  arrow_yend=c(15,15,0,0,0),
  mar=c(1,1,1,0))

dev.off()

#####
# PIM

# Negative mode
peak.list.pimN <- peakList(xr[[9]])

# AcPIM3
peak.frag.PIM3Ac1 <- peakFragPPM(peak.list.pimN, tags=MyMapFrag
  $PIM3Ac1$fragments, ppmCut=10)

found.frag.PIM3Ac1 <- rmNullObs(peak.frag.PIM3Ac1)
# Split the nested list into a single data frame
found.frag.PIM3Ac1 <- unlist(found.frag.PIM3Ac1, recursive=FALSE)
all.frag.PIM3Ac1 <- t(sapply(1:length(found.frag.PIM3Ac1), function(i)
  found.frag.PIM3Ac1[[i]]))

# scans meeting fragmentation criteria, ie same parent, collision energy,
  etc
setList.PIM3Ac1 <- all.frag.PIM3Ac1[(all.frag.PIM3Ac1[,6] >= 1575.934 &
  all.frag.PIM3Ac1[,6] <= 1575.974) # same parent mass
  # & all.frag.acpim3[,6] == 35 # same collision energy
  ,]

```

```

setList.PIM3Ac1

# pdf("~/YM_FigureS7.pdf", family="ArialMT", height=6.83, width=6.83)

fragPlot(as.data.frame(peak.list.pimN[[200]]), limits=c(100, 1725),
  refFrgs=MyMapFrgs$PIM3Ac1$fragments, ppm_labadj=-10,
  ppm_xadj=c(-30,160,-30,-150,110,80),
  ppm_yadj=c(60,190,60,50,55,55),
  arrow_x=c(-65,80,-65,-100,50,0),
  arrow_xend=c(-15,-5,-15,-15,0,-5),
  arrow_y=c(30,160,30,25,30,30),
  arrow_yend=c(5,5,5,5,5,5),
  mar=c(1,1,1,0))

# dev.off()

#####
# Positive mode

peak.list.pimP <- lapply(1:3, function(i) peakList(xr[[i]]))

PIMp.parents <- unlist(MycoMass[grepl("PIM", MycoMass
$Abbreviation),c(3:5)])

PIMdelR2 <- c(551.503387058, 579.534687186, 593.550337251, 1492.09221,
1270.88910, 1671.17159, 1713.21854, 1861.25571)

peak.pimp <- peakFrgsPPM(peak.list.pimP, tags=PIMdelR2, ppmCut=10)
found.frgs.pimp <- rmNullObs(peak.pimp)
found.frgs.pimp <- unlist(found.frgs.pimp, recursive=FALSE)
found.frgs.pimp <- unlist(found.frgs.pimp, recursive=FALSE)
all.frgs.pimp <- t(sapply(1:length(found.frgs.pimp), function(j)
found.frgs.pimp[[j]]))

setList.pimp <- all.frgs.pimp[all.frgs.pimp[,9] <= 2400 &
all.frgs.pimp[,9] >= 1200,]
setList.pimp

PIMp.list <- sapply(1:length(PIMp.parents), function(i)
setList.pimp[setList.pimp[,6] <= PIMp.parents[i] + 0.03 & setList.pimp[,6]
>= PIMp.parents[i] - 0.03,])

PIMp.list <- rmNullObs(PIMp.list)

```



```

# Ac2PIM1 [H+] 1492.092
refFrag1492 <- c(1492.09221, PIMdelR2[1:3])

fragPlot(as.data.frame(peak.list.pimP[[1]][[173]]), limits=c(140, 1525),
refFrag1492)

# AcPIM1 [NH4+] 1270.889
refFrag1270 <- c(1270.88910, PIMdelR2)

fragPlot(as.data.frame(peak.list.pimP[[2]][[237]]), limits=c(140, 1300),
refFrag1270)
# Has AcPIM but also CL! Not useable

# Ac2PIM2 [NH4+] 1671.172
refFrag1671 <- c(1671.17159, PIMdelR2)

fragPlot(as.data.frame(peak.list.pimP[[3]][[161]]), limits=c(140, 1700),
refFrag1671)

# Ac2PIM2 [NH4+] 1713.219
refFrag1713 <- c(1713.21854, PIMdelR2)

fragPlot(as.data.frame(peak.list.pimP[[2]][[303]]), limits=c(140, 1725),
refFrag1713)

#####

# PAs

peak.list.pa <- peakList(xr[[11]])

peak.frag.pa <- peakFragPPM(peak.list.pa, tags=MyMapFrag$PA_673.n
$fragments, ppmCut=10)

found.frag.pa <- rmNullObs(peak.frag.pa)
# Split the nested list into a single data frame

```

```

found.frag.pa <- unlist(found.frag.pa, recursive=FALSE)
all.frag.pa <- t(sapply(1:length(found.frag.pa), function(i)
  found.frag.pa[[i]]))

setList.pa <- all.frag.pa[(all.frag.pa[,6] >= 671 &
  all.frag.pa[,6] <= 674)
  & (all.frag.pa[,9] >= 421 &
  all.frag.pa[,9] <= 479),]
setList.pa

refFrag.pa_671 <- c(as.numeric(MyMapFrag$PA_671.n$parent),
  as.numeric(MyMapFrag$PA_671.n$fragments))
refFrag.pa_673 <- c(as.numeric(MyMapFrag$PA_673.n$parent),
  as.numeric(MyMapFrag$PA_673.n$fragments))

fragPlot(as.data.frame(peak.list.pa[[501]]), limits=c(140, 680),
  refFrag=refFrag.pa_673[-1], ppm_labadj=-120,
  ppm_xadj=c(-15,-70,-60, 50),
  ppm_yadj=c(350,275, 375, 20),
  arrow_x=c(-40,-60,-50,-5),
  arrow_xend=c(-11,-13,-11,-5),
  arrow_y=c(140,90,160,0),
  arrow_yend=c(20,15,20,0),
  mar=c(1,1,1,0))

# dev.off()

# pdf("~/YM_FigureS9.pdf", family="ArialMT", height=6.83, width=6.83)

fragPlot(as.data.frame(peak.list.pa[[497]]), limits=c(140, 680),
  refFrag=refFrag.pa_671[-1], ppm_labadj=-120,
  ppm_xadj=c(-15, -40, 50),
  ppm_yadj=c(350, 400, 20),
  arrow_x=c(-40,-50,-5),
  arrow_xend=c(-11,-11,-5),
  arrow_y=c(140,160,0),
  arrow_yend=c(20,20,0),
  mar=c(1,1,1,0))

# dev.off()

# repeat for all PAs
peak.frag.pa2 <- peakFragPPM(peak.list.pa, tags=MyMapFrag$PA_series.n

```

```
$fragments, ppmCut=10)
```

```
found.frag.pa2 <- rmNullObs(peak.frag.pa2)
```

```
# Split the nested list into a single data frame
```

```
found.frag.pa2 <- unlist(found.frag.pa2, recursive=FALSE)
```

```
all.frag.pa2 <- t(sapply(1:length(found.frag.pa2), function(i)  
  found.frag.pa2[[i]]))
```

```
f.scan <- paste(all.frag.pa2[,1], all.frag.pa2[,2], sep="_")
```

```
mulFrag.pa <- all.frag.pa2[duplicated(f.scan) | duplicated(f.scan,  
  fromLast=TRUE),]
```

```
setList.pa <- mulFrag.pa[(mulFrag.pa[,3] >= 152.9 &  
  mulFrag.pa[,3] <= 153) & (mulFrag.pa[,9] >= 100 &  
  mulFrag.pa[,9] <= 800),]
```

```
setList.pa
```

```
# Looks good. Refine labels
```

```
# 645
```

```
fragPlot(as.data.frame(peak.list.pa[[477]]), limits=c(140, 675),  
  refFrag=as.numeric(MyMapFrag$PA_series.n$fragments))
```

```
# or
```

```
fragPlot(as.data.frame(peak.list.pa[[478]]), limits=c(140, 675),  
  refFrag=as.numeric(MyMapFrag$PA_series.n$fragments))
```

```
# or
```

```
fragPlot(as.data.frame(peak.list.pa[[494]]), limits=c(140, 675),  
  refFrag=as.numeric(MyMapFrag$PA_series.n$fragments))
```

```
# 689
```

```
fragPlot(as.data.frame(peak.list.pa[[541]]), limits=c(140, 700),  
  refFrag=as.numeric(MyMapFrag$PA_series.n$fragments))
```

```
# 675
```

```
fragPlot(as.data.frame(peak.list.pa[[588]]), limits=c(140, 725),  
  refFrag=as.numeric(MyMapFrag$PA_series.n$fragments))
```

```
# or
```

```
fragPlot(as.data.frame(peak.list.pa[[657]]), limits=c(140, 725),  
  refFrag=as.numeric(MyMapFrag$PA_series.n$fragments))
```

```
#####  
# What about TAGs?
```

```
peak.list.tag <- peakList(xr[[10]])
```

```
TAGs <- c(551.5034, 565.5190, 579.5347, 593.5503, 607.5660, 621.5816,  
635.5973, 549.4878, 563.5033, 577.5190, 591.5346, 605.5503, 619.5659,  
633.5816, 547.4721, 561.4877, 575.5034, 589.5190, 603.5347, 617.5503,  
631.5660)
```

```
peak.frag.TAGs <- peakFragPPM(peak.list.tag, tags=TAGs, ppmCut=10)
```

```
found.frag.TAGs <- rmNullObs(peak.frag.TAGs)  
# Split the nested list into a single data frame  
found.frag.TAGs <- unlist(found.frag.TAGs, recursive=FALSE)  
all.frag.TAGs <- t(sapply(1:length(found.frag.TAGs), function(i)  
  found.frag.TAGs[[i]]))
```

```
f.scan.tag <- paste(all.frag.TAGs[,1], all.frag.TAGs[,2], sep="_")  
mulFrag.tag <- all.frag.TAGs[duplicated(f.scan.tag) |  
  duplicated(f.scan.tag, fromLast=TRUE),]
```

```
mulFrag.tag
```

```
# TAG 850  
refFrag.tag_851 <- c(as.numeric(MyMapFrag$TAG_851.p$fragments))
```

```
# Use this one  
fragPlot(as.data.frame(peak.list.tag[[112]]), limits=c(500, 900),  
  refFrag=TAGs, ppmThresh=15, mar=c(1,1,1,0))
```

```
# TAG 864  
refFrag.tag_864 <- c(as.numeric(MyMapFrag$TAG_865.p$fragments))
```

```
# Use this one  
fragPlot(as.data.frame(peak.list.tag[[91]]), limits=c(500, 950),  
  refFrag=TAGs, ppmThresh=15, mar=c(1,1,1,0))
```

```
# TAG 878  
refFrag.tag_879 <- c(as.numeric(MyMapFrag$TAG_879.p$fragments))
```

```

# Use this one
fragPlot(as.data.frame(peak.list.tag[[160]]), limits=c(500, 950),
refFragS=TAGs, ppmThresh=15, mar=c(1,1,1,0))

# TAG 878
refFragS.tag_879 <- c(577.5190, 579.5347, 605.5503)

# pdf("~/Documents/Projects/Moody/Yasu/YM_FigureS2E_v2_raw.pdf",
family="ArialMT", height=6.83, width=6.83)

fragPlot(as.data.frame(peak.list.tag[[160]]), limits=c(500, 925),
refFragS=refFragS.tag_879, ppm_labadj=5,
ppm_xadj=c(-70,55,30),
ppm_yadj=c(35,-15,35),
arrow_x=c(-65,0,5),
arrow_xend=c(-15,0,0),
arrow_y=c(30,0,30),
arrow_yend=c(5,0,5),
mar=c(1,1,1,0))

dev.off()

#####

# TAG 906
refFragS.tag_907 <- c(as.numeric(MyMapFragS$TAG_907.p$fragments))

# Use this one
fragPlot(as.data.frame(peak.list.tag[[94]]), limits=c(500, 950),
refFragS=TAGs, ppmThresh=10, mar=c(1,1,1,0))

# fragPlot(as.data.frame(peak.list.1110[[6]][[169]]), limits=c(500, 950),
refFragS=TAGs, ppmThresh=20, mar=c(1,1,1,0))

#####
# DAGs

peak.list.dag <- peakList(xr[[8]])

```

```

DAG.parents <- MycoMass[grepl("DAG", MycoMass$Abbreviation),4][36:56]

DAGdelOH <- DAG.parents-35.03711
DAGdelR <- t(sapply(1:length(DAG.parents), function(i) DAG.parents[i]-
  MyMapFragments$PA_series.n$fragments[c(8:10, 13:18)]-18.033825553))

DAGs <- data.frame(Parent=DAG.parents, DAGdelOH=DAGdelOH, DAGdelR)

# fragments from DAGs

DAG.List <- lapply(1:length(DAGs[,1]), function(i) {
  peak.frag.DAG <- peakFragPPM(peak.list.dag,
  tags=as.numeric(DAGs[i,]), ppmCut=10)
  found.frag.DAG <- rmNullObs(peak.frag.DAG)
# Split the nested list into a single data frame
  found.frag.DAG <- unlist(found.frag.DAG, recursive=FALSE)
  all.frag.DAG <- t(sapply(1:length(found.frag.DAG), function(j)
  found.frag.DAG[[j]]))
  all.frag.DAG
})

fDAG.list <- lapply(1:length(DAG.List), function(i) {
  setList.DAG <- DAG.List[[i]][DAG.List[[i]][,9] <= 600 & DAG.List[[i]][,
  6] <= DAGs[i,1] + 0.01 & DAG.List[[i]][,6] >= DAGs[i,1] - 0.01,]
  setList.DAG
})
fDAG.list

# DAG 612
refFrag.dag612 <- as.numeric(DAGs[8,])[-1]

# Use this one
fragPlot(as.data.frame(peak.list.dag[97]), limits=c(140, 650),
  refFrag=refFrag.dag612, ppmThresh=15, mar=c(1,1,1,0))

```