

Supplementary materials for “Relationships of gut microbiota, short-chain fatty acids, inflammation, and the gut barrier in Parkinson’s disease”

Supplementary methods	1
Short-chain fatty acid measurements	1
Inflammatory and permeability marker measurements	2
Statistical analyses.....	3
Supplementary figures	3
Figure S1. Correlations for variables of interest in full data and subsets.	3
Figure S2. Scatterplots for significantly correlated clinical variables and SCFAs or inflammatory markers. ..	5
Figure S3. Correlations of inverse Simpson and Shannon diversity for SCFAs and inflammatory markers. ...	6
Figure S4. NMDS ordination plots for beta diversity, control/PD status, and the stool markers and SCFAs associated with the most notable beta diversity differences.....	7
Figure S5. Butyric acid, stool NGAL, and zonulin in different enterotypes among PD patients and control subjects.....	8
Figure S6. Scatter plots of differentially abundant genera for SCFAs.....	9
Figure S7. Scatter plots of differentially abundant genera for stool inflammatory markers.....	10
Supplementary tables.....	11
Table S1. Marker and SCFA concentrations by PD/control status and sex.....	11
Table S2. Linear regression for alpha diversity, inflammatory and permeability markers/SCFAs, and PD/control status.....	12
Table S3. Beta diversity, inflammatory and permeability markers/SCFAs and PD status.....	13
Table S4. Differences in inflammatory and permeability marker/SCFA levels between enterotypes.....	14
Table S5. Differentially abundant taxa for variables of interest.	15

Supplementary methods

Short-chain fatty acid measurements

Detection and quantification of SCFAs in the stool samples were performed at the Norwegian University of Life Sciences using a Trace 1310 gas chromatograph (Thermo Fisher Scientific, USA) equipped with an autosampler, a split/splitless injector, a flame ionization detector, and a Stabilwax DA column (30 m, 0.25 mm ID, 0.25 µm film; Thermo Fisher Scientific, USA). Samples were suspended in MilliQ water (1:1 w/v) and subsequently homogenized by vortexing using glass beads (size: 2.5-3.5 mm) in the solution. The samples were then diluted with 0.2% formic acid and 500 µM of an internal standard (2-methylvaleric acid; 1:1 w/v) before a 5 min centrifugation step at 13 000 rpm. The internal standard was included to control equal sampling volume and was used for normalization of the samples. 300 µl of the supernatant was applied to spin columns (0.2 µm filter; VWR, USA) and centrifuged at 10 000 rpm for 5 min. The eluate was transferred to 300 µl GC vials (VWR) and analyzed in the GC-FID instrument. A standard solution was made with final concentrations of 0.1% formic acid and 1000 µM of each of the acids (acetic acid, propionic acid, butyric

acid, isobutyric acid, valeric acid, and 2-methylvaleric acid). All acids were purchased from Sigma-Aldrich, Germany. Samples were injected into the GC system in volumes of 0.2 μ L. Split mode was set to 2.5:200, and injector and detector temperatures were set to 250 and 275 $^{\circ}$ C respectively. Helium was used as a carrier gas. Temperature intervals comprised a 2-min step at 90 $^{\circ}$ C followed by a 6-min increase to 150 $^{\circ}$ C and a 2-min increase to 245 $^{\circ}$ C followed by a hold at this temperature for 4.9 min. The software Chromeleon 7.2 (Thermo Fisher Scientific) was used for instrument control, quantification, and data analysis.

Inflammatory and permeability marker measurements

To prepare stool for the Zonulin Stool ELISA (ALPCO catalog # 30-ZONHU-E01) kit, the dilution/extraction buffer was prepared according to the manufacturer's instructions and added to thawed raw stool samples at a ratio of 1 mg stool to 50 mL buffer. The assay was then performed according to the manufacturer's protocol. Reference absorbance values (absorbance at 620 nm) were subtracted from absorbance values at 450 nm, and zonulin concentrations were interpolated from a standard curve using a quadratic regression according to manufacturer's recommendations and multiplied by the dilution factor (50) to obtain actual concentrations in the stool (ng/mg).

For measurements of stool calprotectin, neutrophil gelatinase-associated lipocalin (NGAL), and cytokines, frozen stool samples were suspended in homogenization buffer (125 mM Tris, 15 mM MgCl₂, 2.5 mM EDTA pH 7.2, 1% Triton X-100, 1 tablet protease inhibitors [1697498, Roche] per 10 mL buffer) and homogenized in a TissueLyser-II (QIAGEN, 20 Hz) with a 5 mm metal bead (QIAGEN). Debris were pelleted, and supernatants were collected. The total protein concentration in each supernatant was measured by BCA assay (Pierce) according to the manufacturer's protocol, and samples were diluted to 6 mg/mL in homogenization buffer.

For the LEGEND MAX™ Human MRP8/14 (Calprotectin) ELISA Kit (Biolegend catalog # 439707), 6 mg/mL stool samples were diluted 1:20 in assay buffer A, and 50 μ L (15 μ g protein) were loaded into wells. The assay was performed according to the manufacturer's protocol. Reference absorbance values (absorbance at 570 nm) were subtracted from absorbance values at 450 nm, and calprotectin concentrations were interpolated from a standard curve using a quadratic regression according to manufacturer's recommendations and multiplied by the dilution factor (20) to obtain actual concentrations in the stool lysate (ng/mL).

Stool NGAL levels were measured using the LEGEND MAX™ Human NGAL (Lipocalin-2) ELISA Kit (Biolegend catalog # 443407). 6 mg/mL stool samples were diluted 1:50 in assay buffer B, and 50 μ L (6 μ g protein) were loaded into wells. The assay was performed according to the manufacturer's protocol and analyzed in the same way as calprotectin.

The cytokines interferon gamma (IFN γ), interleukin (IL)-1 β , IL-2, IL-4, IL-6, CXCL8, IL-10, IL-12p70, IL-13, and tumor necrosis factor (TNF) were measured in stool and plasma by multiplexed immunoassay using the V-PLEX Proinflammatory Panel 1 Human Kit (Meso Scale Discovery, MSD, Rockville, MD, catalog # K15049D). 6 mg/mL stool lysates were diluted 1:1 in diluent 2, and 50 μ L (150 μ g) of each sample were loaded. Fifty microliters of undiluted plasma were loaded for each test. The multiplexed immunoassay was performed according to the manufacturer's protocol using the MSD QuickPlex instrument, and the results were evaluated on the MSD software platform. Resulting values were multiplied by the dilution factor to obtain actual concentrations in the stool lysates and plasma (pg/mL).

Lipopolysaccharide binding protein (LBP) concentrations (μ g/mL) were measured in plasma using the Human LBP Kit (MSD catalog # K151IYC) on the MSD platform according to the manufacturer's protocol.

All samples were run in duplicate in each assay, and roughly equal numbers of control and PD subject samples were included on each plate. In the multiplexed immunoassays, stool and plasma samples from each subject were run on the same plate. No systematic differences were observed in values obtained from different runs.

Statistical analyses

We performed all statistical analyses in R,³² and managed microbiota data with the phyloseq package.³³ The full analysis workflow is available as a supplementary file (Supplementary R Markdown). For statistical comparisons between SCFAs/markers, clinical variables, enterotypes, and alpha diversity, we used the Kruskal-Wallis test, Wilcoxon rank sum test, or Pearson correlations depending on the types of variables. When multiple comparison corrections were included, we used the Benjamini & Hochberg false discovery rate, correcting SCFAs, stool markers, and plasma markers separately. Principal Component Analysis (PCA) was used to merge intercorrelated inflammatory markers.

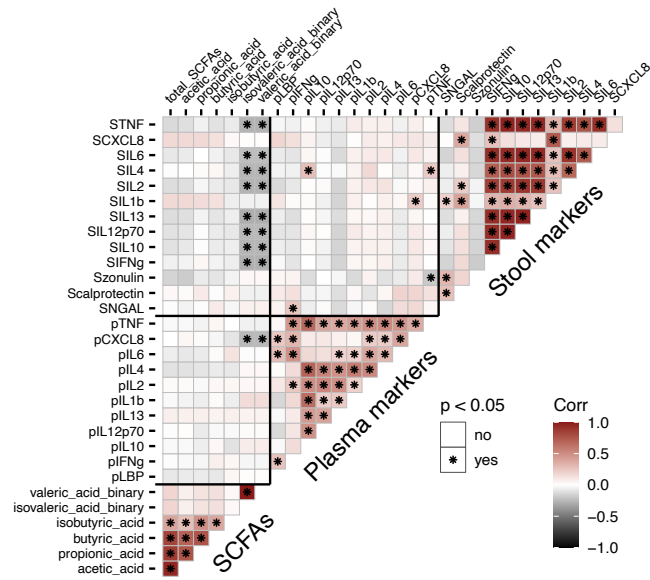
In all microbiome-related comparisons, we tested the full data (patients and control subjects together), and PD-only and control-only subsets. Additionally, we ran confounder-corrected comparisons specific to each test. For alpha and beta diversity comparisons, we used OTU-level data subsampled to the lowest number of reads in a sample (7165). Alpha diversity comparisons included two measures: Shannon and inverse Simpson (both indices include richness and evenness); after preliminary screening with correlations, we explored the most interesting cases with linear regression - corrected for PD/control and sex - with the SCFA or marker variables log-transformed to address heteroskedasticity. Beta diversity was evaluated with PERMANOVAs on Bray-Curtis distance matrices, run with the *adonis* (univariable) and *adonis2* (multivariable) tests from the *vegan* package³⁴, and visualized with Non-Metric Multidimensional scaling (NMDS). The variables in confounder-corrected *adonis2* models were sex, Rome III 9-15 sum score, and BMI. For differential abundance testing, we used DESeq2³⁵ on trimmed data (genera and families with > 5 reads in > 50% of the samples), and ran four models: full data, PD-only, control-only, and full data corrected for PD/control status, Rome III 9-15 sum score, and sex.

Supplementary figures

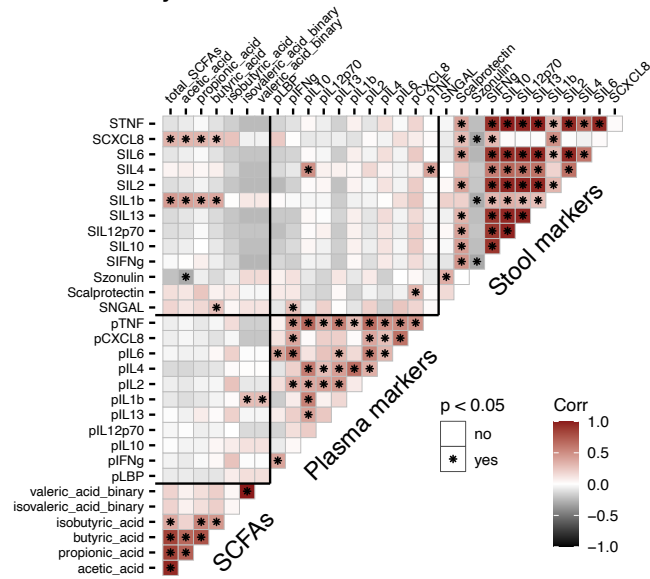
Figure S1. Correlations for variables of interest in full data and subsets.

* : $p < 0.05$.

A. Full data



B. Control only



C. PD only

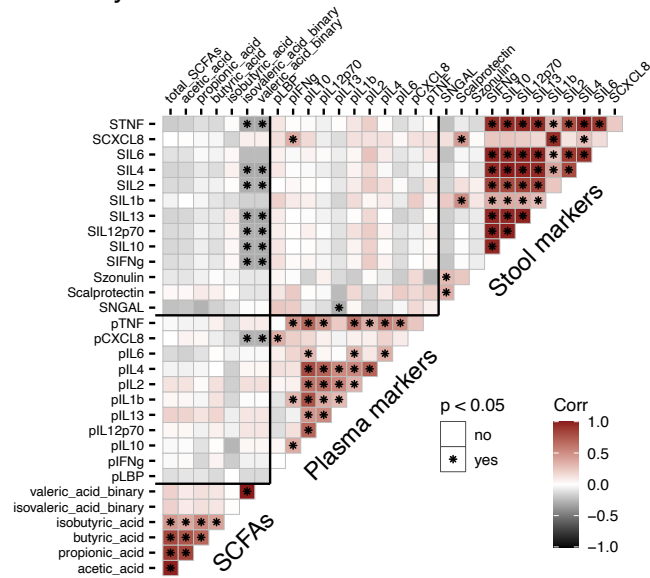
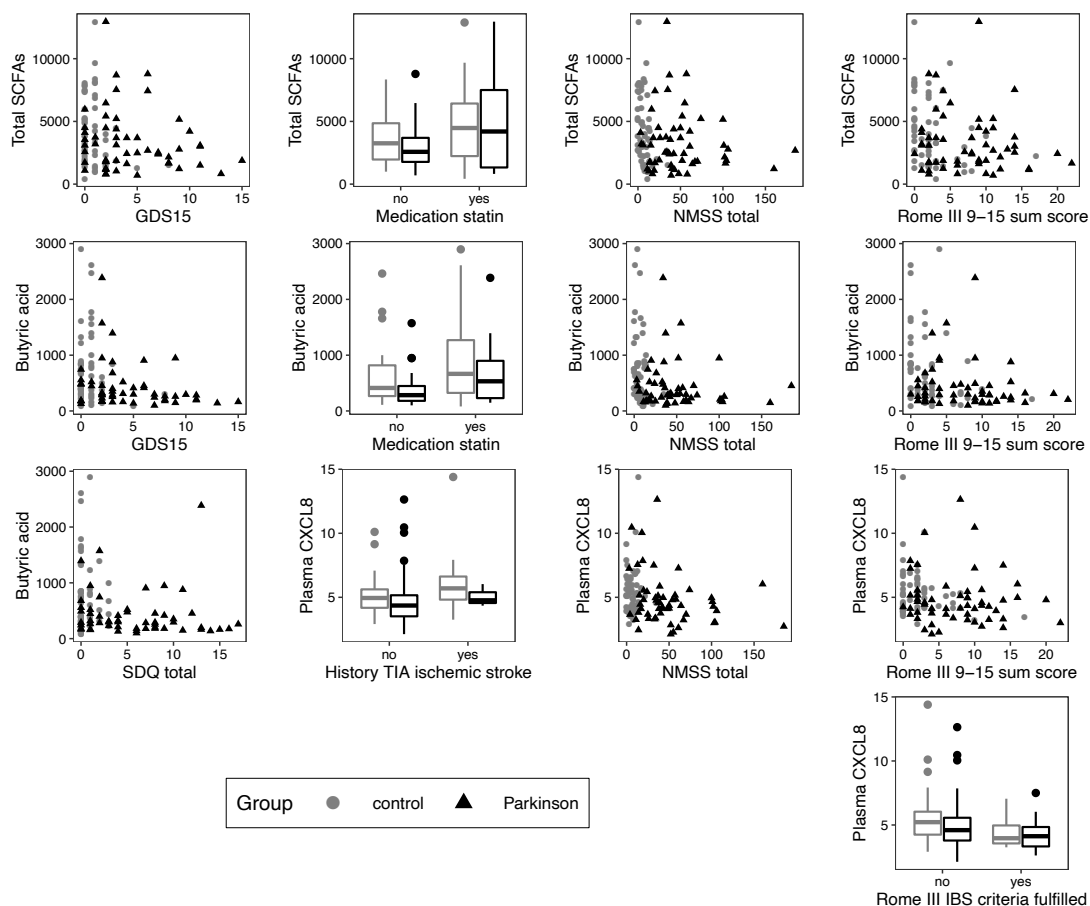


Figure S2. Scatterplots for significantly correlated clinical variables and SCFAs or inflammatory markers.

A. Variables measured in all subjects; B. Variables measured only in PD patients.

A



B

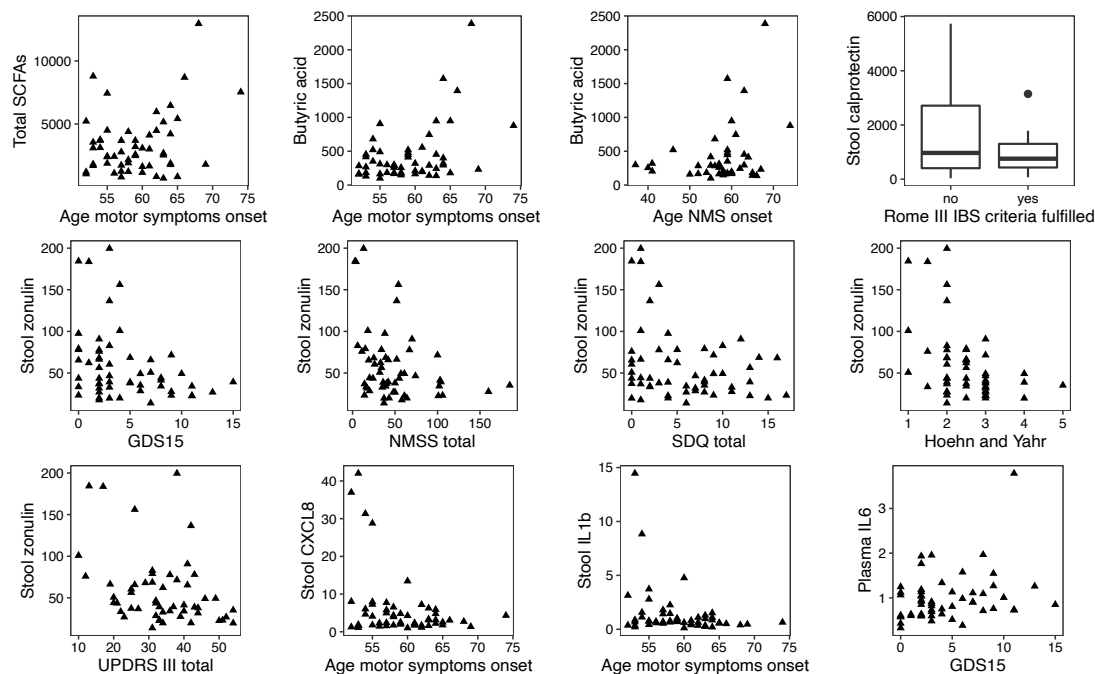
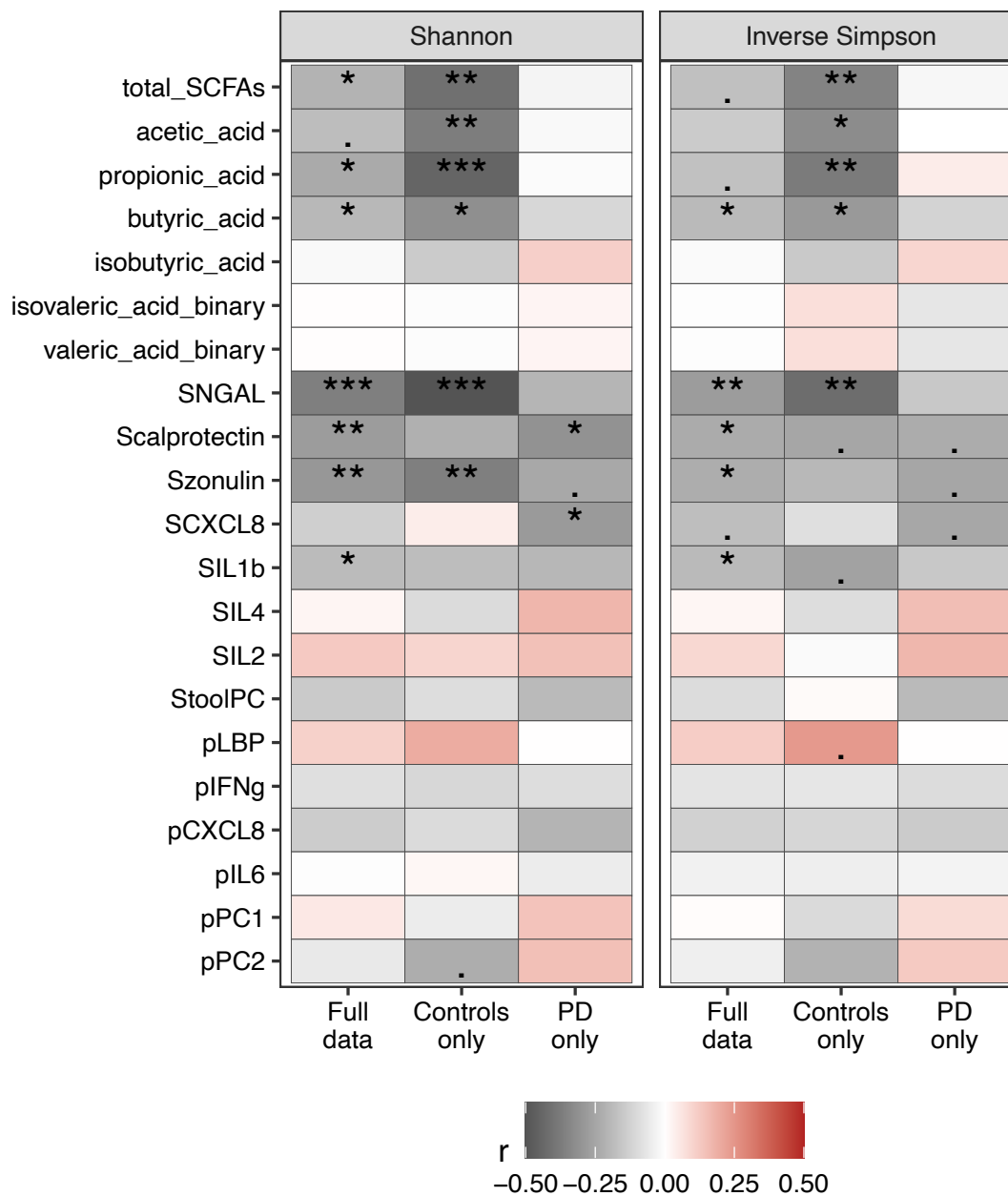


Figure S3. Correlations of inverse Simpson and Shannon diversity for SCFAs and inflammatory markers.



Significance level is marked as follows: *** : $p \leq 0.001$ / ** : $p \leq 0.01$ / * : $p \leq 0.05$ / . : $0.1 > p > 0.05$.

Figure S4. NMDS ordination plots for beta diversity, control/PD status, and the stool markers and SCFAs associated with the most notable beta diversity differences.

Variables were split into two categories by median.

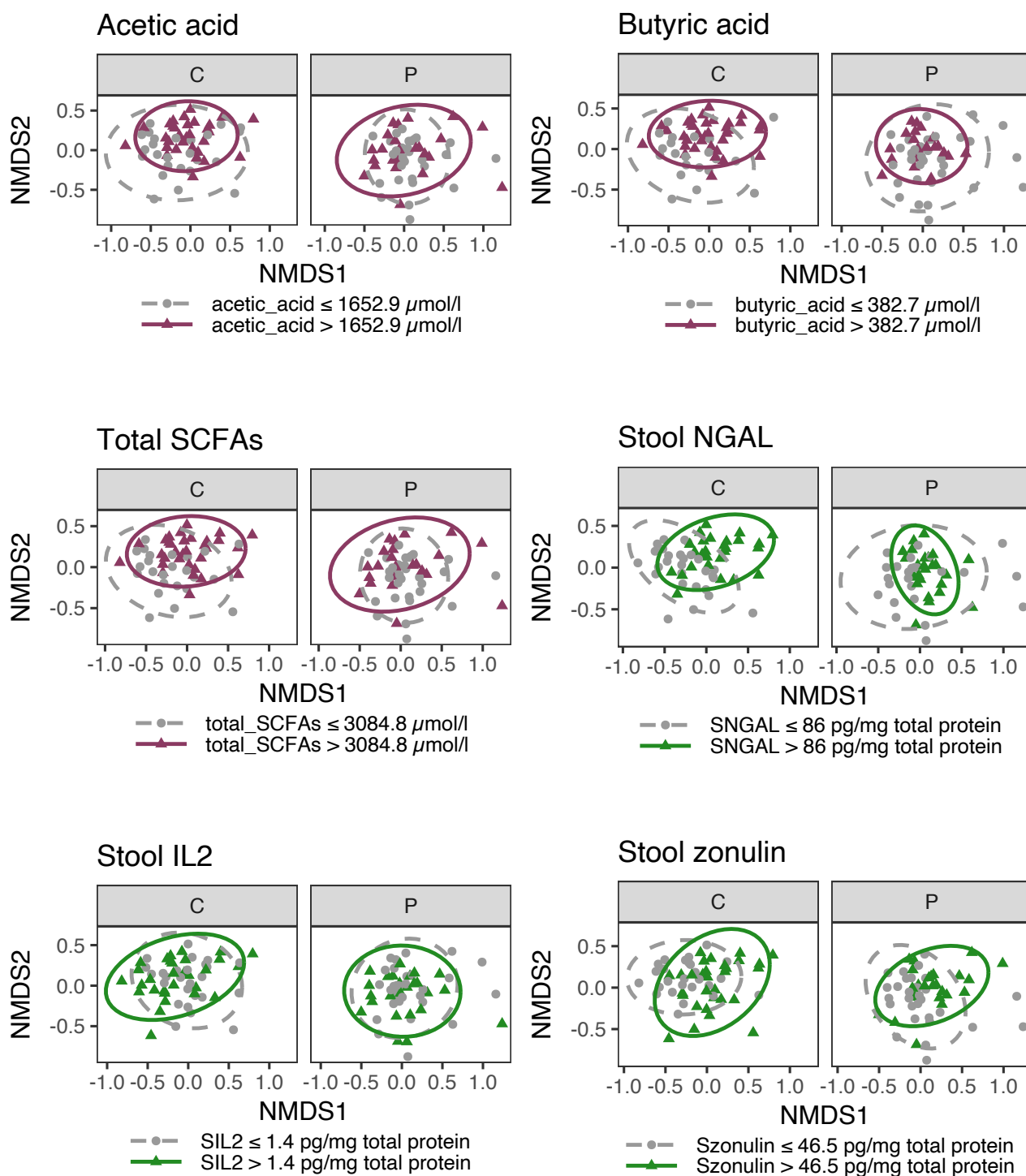


Figure S5. Butyric acid, stool NGAL, and zonulin in different enterotypes among PD patients and control subjects.

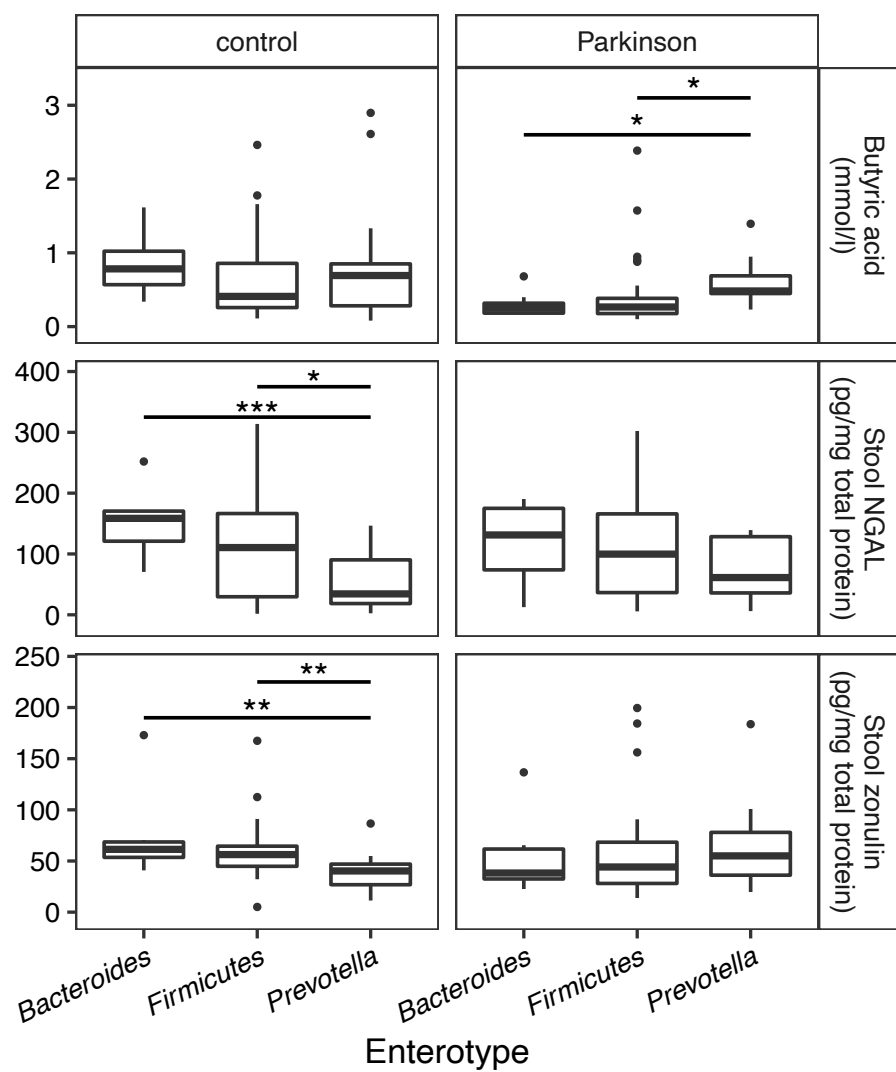


Figure S6. Scatter plots of differentially abundant genera for SCFAs.

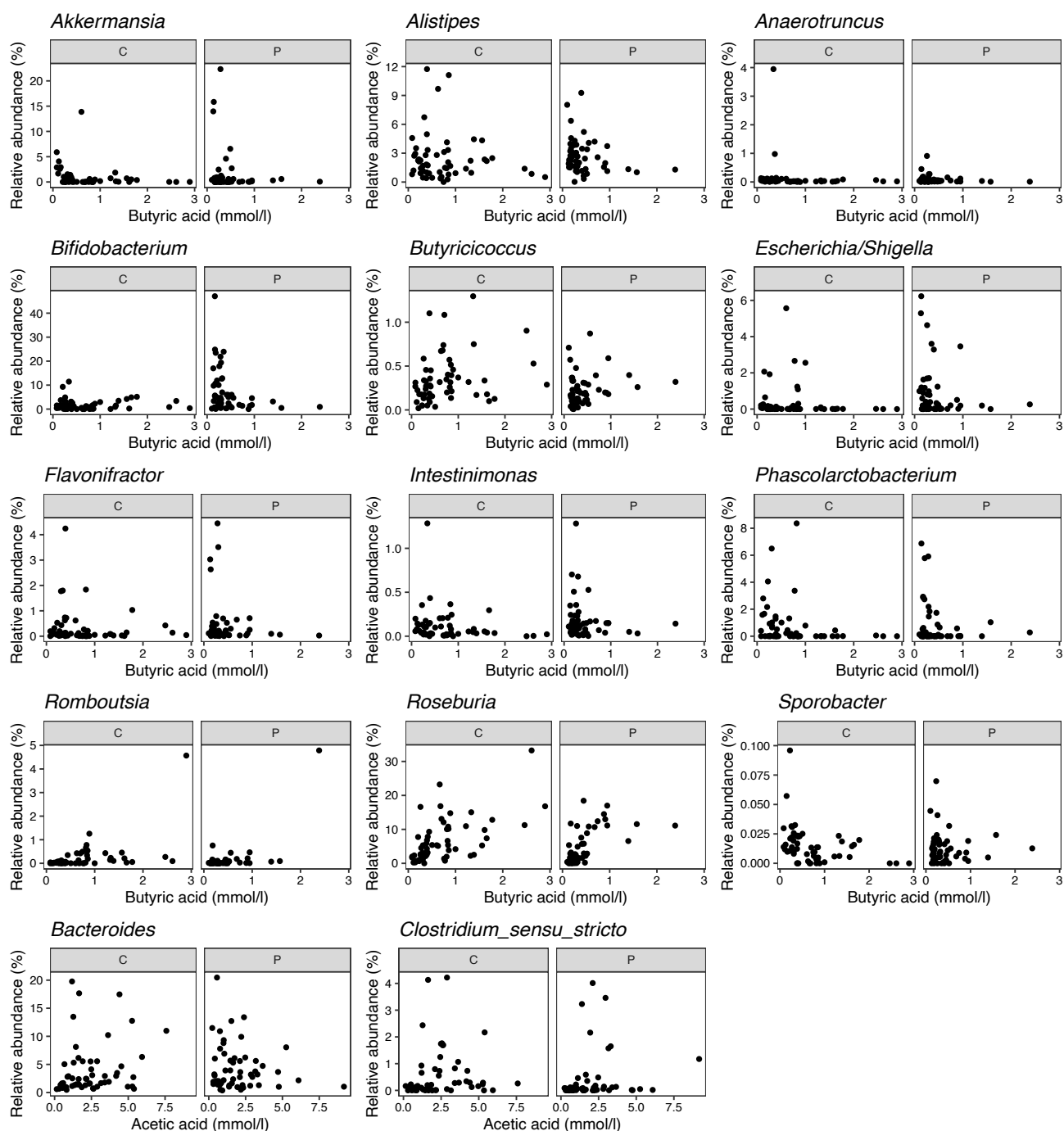
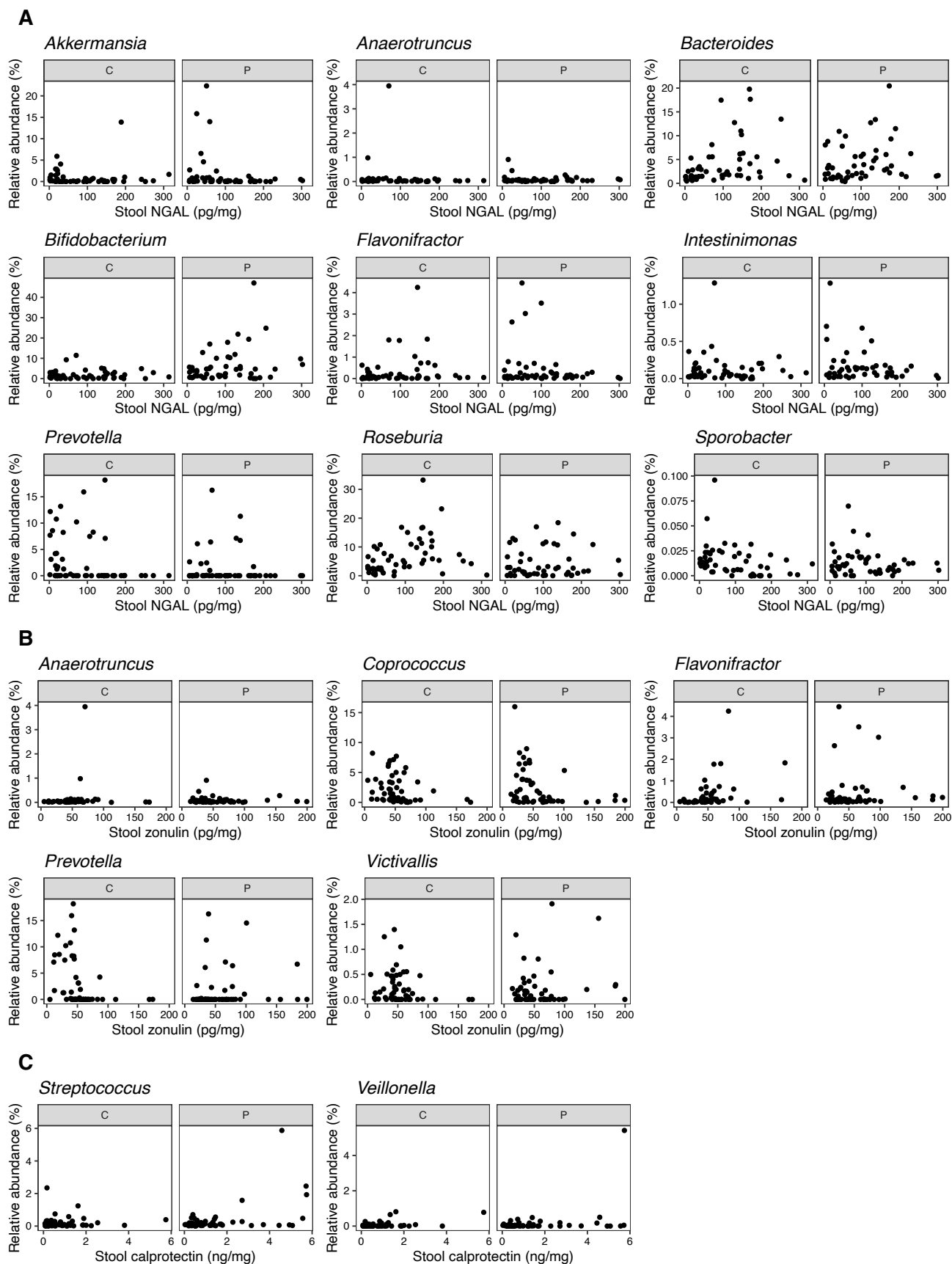


Figure S7. Scatter plots of differentially abundant genera for stool inflammatory markers.

A. NGAL; B. Zonulin; C. Calprotectin.



Supplementary tables

Table S1. Marker and SCFA concentrations by PD/control status and sex

Variable	Unit	Full data			Females			Males		
		Control subjects	PD patients	p-value	Control subjects	PD patients	p-value	Control subjects	PD patients	p-value
Total SCFAs	μmol/l	4209.14 ± 2688.93	3288.72 ± 2388.22	0.051	3670.79 ± 2214.48	3301.34 ± 2513.4	0.335	4787.37 ± 3056.69	3277.41 ± 2314.98	0.053
Acetic acid	μmol/l	2390.65 ± 1699.99	1938.61 ± 1669.64	0.092	2012.88 ± 1374.1	1965.12 ± 1849.39	0.487	2796.4 ± 1936.46	1914.84 ± 1523.55	0.064
Propionic acid	μmol/l	840.13 ± 509.61	644.38 ± 351.39	0.042	781.83 ± 480.18	640.39 ± 283.66	0.388	902.75 ± 541.49	647.95 ± 407.75	0.057
Butyric acid	μmol/l	736.26 ± 633.78	424.61 ± 405.92	0.001	658.33 ± 661.11	436.15 ± 461.58	0.101	819.95 ± 604.11	414.26 ± 356.8	0.003
Isobutyric acid	μmol/l	123.99 ± 43.24	131.02 ± 46.37	0.615	119.63 ± 34.52	134.22 ± 49.94	0.411	128.88 ± 51.61	128.05 ± 43.56	0.870
Isovaleric acid	μmol/l	65.59 ± 125.07	92.63 ± 149.41	0.401	51.89 ± 96.79	72.88 ± 131.65	0.723	80.31 ± 150.22	110.33 ± 163.98	0.508
Valeric acid	μmol/l	60.24 ± 113.12	65.82 ± 99.15	0.534	50.35 ± 105.8	60.05 ± 97.72	0.583	71.26 ± 121.91	70.8 ± 101.83	0.880
Stool NGAL	pg/mg total protein	93.88 ± 78.46	100.92 ± 74.89	0.531	97.4 ± 76.32	97.77 ± 72.12	0.908	89.95 ± 82.12	103.72 ± 78.53	0.391
Stool calprotectin	pg/mg total protein	872.01 ± 1000.65	1499.67 ± 1616.72	0.023	665.37 ± 772.91	1502.36 ± 1696.54	0.019	1102.49 ± 1178.53	1497.26 ± 1571.93	0.489
Stool zonulin	pg/mg total protein	52.86 ± 30.38	58.53 ± 42.73	0.807	56.18 ± 29.02	53.03 ± 31.41	0.436	49.3 ± 31.93	63.47 ± 50.87	0.603
Stool IFNγ	pg/mg total protein	6.35 ± 5.5	5.68 ± 4.8	0.542	6.41 ± 4.01	4.85 ± 3.83	0.078	6.27 ± 6.83	6.42 ± 5.49	0.282
Stool IL10	pg/mg total protein	0.73 ± 0.73	0.67 ± 0.69	0.485	0.76 ± 0.49	0.6 ± 0.61	0.090	0.71 ± 0.94	0.73 ± 0.77	0.353
Stool IL12p70	pg/mg total protein	1.36 ± 1.17	1.13 ± 1.06	0.178	1.44 ± 0.86	1.02 ± 0.89	0.040	1.27 ± 1.45	1.22 ± 1.19	0.558
Stool IL13	pg/mg total protein	6.26 ± 4.32	5.73 ± 3.99	0.477	6.73 ± 3.63	5.15 ± 3.86	0.065	5.76 ± 4.98	6.24 ± 4.11	0.233
Stool IL1b	pg/mg total protein	1.24 ± 1.53	1.38 ± 2.28	0.505	0.96 ± 0.52	1.46 ± 2.92	0.132	1.55 ± 2.12	1.3 ± 1.6	0.757
Stool IL2	pg/mg total protein	1.98 ± 1.69	1.9 ± 1.88	0.481	2.09 ± 1.23	1.52 ± 1.24	0.070	1.85 ± 2.09	2.24 ± 2.27	0.312
Stool IL4	pg/mg total protein	0.31 ± 0.34	0.25 ± 0.24	0.282	0.29 ± 0.18	0.23 ± 0.23	0.078	0.33 ± 0.45	0.26 ± 0.25	0.591
Stool IL6	pg/mg total protein	1.29 ± 1.18	1.15 ± 1.05	0.378	1.35 ± 0.9	0.94 ± 0.8	0.047	1.23 ± 1.44	1.34 ± 1.22	0.304
Stool CXCL8	pg/mg total protein	5.82 ± 7.53	5.96 ± 8.6	0.530	4.23 ± 3.66	6.96 ± 11.14	0.272	7.53 ± 9.98	5.06 ± 5.49	0.757
Stool TNF	pg/mg total protein	1.63 ± 1.56	1.54 ± 1.51	0.643	1.75 ± 1.2	1.5 ± 1.5	0.171	1.51 ± 1.88	1.57 ± 1.54	0.247
Plasma LBP	pg/mg total protein	3.38 ± 1.45	3.21 ± 1.26	0.546	3.15 ± 1.43	3.08 ± 1.02	0.861	3.62 ± 1.46	3.33 ± 1.45	0.434
Plasma IFNγ	pg/mg total protein	8.04 ± 6.33	6.77 ± 4.04	0.251	8.9 ± 6.74	7.94 ± 4.56	0.744	7.11 ± 5.86	5.73 ± 3.26	0.240
Plasma IL10	pg/mg total protein	0.45 ± 0.31	0.45 ± 0.3	0.798	0.47 ± 0.3	0.44 ± 0.28	0.646	0.43 ± 0.33	0.46 ± 0.33	1.000
Plasma IL12p70	pg/mg total protein	0.46 ± 0.53	0.58 ± 0.82	0.264	0.55 ± 0.65	0.52 ± 0.53	0.900	0.35 ± 0.33	0.63 ± 1.03	0.101
Plasma IL13	pg/mg total protein	5.35 ± 5.64	6.48 ± 5.99	0.119	7.03 ± 6.64	6.53 ± 6.78	0.796	3.55 ± 3.65	6.44 ± 5.3	0.006
Plasma IL1b	pg/mg total protein	0.23 ± 0.38	0.21 ± 0.31	0.906	0.35 ± 0.49	0.25 ± 0.38	0.484	0.09 ± 0.08	0.17 ± 0.23	0.317
Plasma IL2	pg/mg total protein	0.56 ± 0.64	0.56 ± 0.61	0.652	0.59 ± 0.64	0.49 ± 0.59	0.953	0.53 ± 0.65	0.63 ± 0.62	0.494
Plasma IL4	pg/mg total protein	0.06 ± 0.05	0.07 ± 0.06	0.375	0.08 ± 0.06	0.07 ± 0.06	0.498	0.04 ± 0.03	0.07 ± 0.06	0.057
Plasma IL6	pg/mg total protein	0.96 ± 1.07	0.99 ± 0.55	0.182	0.93 ± 0.36	1.09 ± 0.68	0.707	0.99 ± 1.52	0.89 ± 0.38	0.101
Plasma CXCL8	pg/mg total protein	5.46 ± 1.87	4.81 ± 2.03	0.008	5.25 ± 1.37	4.67 ± 1.94	0.017	5.68 ± 2.29	4.95 ± 2.13	0.145
Plasma TNF	pg/mg total protein	3.53 ± 1.23	3.25 ± 1.31	0.112	3.61 ± 0.95	3.38 ± 1.35	0.205	3.45 ± 1.49	3.14 ± 1.28	0.464

Table S2. Linear regression for alpha diversity, inflammatory and permeability markers/SCFAs, and PD/control status.

A. Model: log(variable) ~ sex + PD/control + Diversity

Variable	Diversity index	p-value			
		Entire model	Sex	PD/control	Diversity
Total SCFAs	Shannon	0.010	0.311	0.025	0.009
Total SCFAs	InvSimpson	0.036	0.349	0.041	0.043
Propionic acid	Shannon	0.006	0.598	0.020	0.004
Propionic acid	InvSimpson	0.045	0.663	0.037	0.050
Butyric acid	Shannon	0.000	0.256	0.000	0.006
Butyric acid	InvSimpson	0.001	0.286	0.001	0.018
Stool NGAL	Shannon	0.000	0.995	0.365	0.000
Stool NGAL	InvSimpson	0.002	0.879	0.256	0.000
Stool calprotectin	Shannon	0.000	0.021	0.062	0.000
Stool calprotectin	InvSimpson	0.000	0.023	0.033	0.000
Stool zonulin	Shannon	0.010	0.883	0.823	0.001
Stool zonulin	InvSimpson	0.046	0.830	0.650	0.006
Stool IL1b	Shannon	0.407	0.130	0.481	0.564
Stool IL1b	InvSimpson	0.258	0.125	0.500	0.229

B. Model: log(variable) ~ sex + PD/control * Diversity

Variable	Diversity index	p-value				
		Entire model	Sex	PD/control	Diversity	Interaction: PD/control and diversity
Total SCFAs	Shannon	0.004	0.346	0.029	0.001	0.045
Total SCFAs	InvSimpson	0.016	0.391	0.014	0.006	0.057
Propionic acid	Shannon	0.001	0.660	0.012	0.000	0.020
Propionic acid	InvSimpson	0.008	0.740	0.003	0.002	0.017
Butyric acid	Shannon	0.000	0.283	0.047	0.002	0.094
Butyric acid	InvSimpson	0.001	0.315	0.011	0.009	0.142
Stool NGAL	Shannon	0.000	0.909	0.092	0.000	0.078
Stool NGAL	InvSimpson	0.001	0.802	0.148	0.000	0.056
Stool calprotectin	Shannon	0.000	0.020	0.371	0.013	0.460
Stool calprotectin	InvSimpson	0.000	0.024	0.491	0.007	0.987
Stool zonulin	Shannon	0.024	0.890	0.839	0.027	0.853
Stool zonulin	InvSimpson	0.078	0.855	0.433	0.206	0.501
Stool IL1b	Shannon	0.558	0.130	0.789	0.855	0.743
Stool IL1b	InvSimpson	0.386	0.130	0.570	0.291	0.712

C. Stool calprotectin and additional confounders (selected based on Fig 2C)

	Simple model	All confounders	Main confounders
Sex	-1.461 ** (0.546)	-1.261 * (0.531)	-1.336 * (0.532)
PD vs control	0.381 * (0.179)	0.108 (0.216)	0.299 · (0.176)
Inverse Simpson index	-0.095 *** (0.016)	-0.088 *** (0.016)	-0.090 *** (0.016)
Diabetes		0.595 * (0.282)	0.622 * (0.283)
Relative with PD		0.461 · (0.242)	0.458 · (0.243)
RBDSQ score		0.057 (0.037)	
sex and inverse Simpson interaction	0.093 *** (0.025)	0.084 *** (0.025)	0.087 *** (0.025)
N	109	109	109
R2	0.307	0.374	0.36
logLik	-144.517	-138.938	-140.169
AIC	301.033	295.876	296.339

*** p < 0.001; ** p < 0.01; * p < 0.05; · p < 0.1. Results given as: Estimate / p-value stars / (Std. Error)

D. Propionic acid and additional confounders (selected based on Fig 2A)

	Simple model	All confounders	Main confounders
Sex	0.048 (0.109)	-0.025 (0.111)	
PD vs control	-3.313 * (1.292)	-2.793 * (1.299)	-3.332 ** (1.266)
Shannon index	-0.925 *** (0.243)	-0.854 *** (0.238)	-0.898 *** (0.239)
GDS15 score		-0.029 (0.023)	
ACE-AT1 medication		0.165 (0.116)	
Statin medication		0.163 (0.128)	
NMSS total score		0.002 (0.003)	
Rome III 9-15 sum score		-0.031 * (0.014)	-0.026 * (0.013)
PD/C and Shannon interaction	0.812 * (0.343)	0.736 * (0.348)	0.853 * (0.336)
N	110	110	110
R2	0.156	0.237	0.188
logLik	-91.635	-86.098	-89.501
AIC	195.271	194.196	191.003

*** p < 0.001; ** p < 0.01; * p < 0.05; · p < 0.1. Results given as: Estimate / p-value stars / (Std. Error)

Table S3. Beta diversity, inflammatory and permeability markers/SCFAs and PD status.

A. Without interactions (all p-values for the variable of interest)

Variable	p (full data)	p (control only)	p (PD only)	p (PD/C corrected)	p (PD/C + confounder* corrected)
Total SCFAs	0.000	0.000	0.066	0.000	0.000
Acetic acid	0.000	0.000	0.275	0.000	0.001
Propionic acid	0.000	0.001	0.217	0.000	0.001
Butyric acid	0.000	0.000	0.004	0.000	0.000
Isobutyric acid	0.591	0.806	0.159	0.430	0.506
Isovaleric acid binary	0.134	0.060	0.362	0.099	0.099
Valeric acid binary	0.134	0.060	0.362	0.099	0.099
Stool NGAL	0.008	0.004	0.307	0.007	0.005
Stool calprotectin	0.058	0.447	0.171	0.103	0.079
Stool zonulin	0.001	0.008	0.021	0.000	0.002
Stool CXCL8	0.164	0.690	0.152	0.152	0.172
Stool IL1b	0.332	0.366	0.401	0.331	0.400
Stool IL4	0.561	0.921	0.088	0.686	0.651
Stool IL2	0.030	0.479	0.076	0.030	0.030
Stool PC	0.016	0.598	0.054	0.022	0.023
Plasma LBP	0.981	0.920	0.894	0.980	0.944
Plasma IFNg	0.900	0.739	0.897	0.894	0.872
Plasma CXCL8	0.566	0.563	0.811	0.676	0.718
Plasma IL6	0.422	0.237	0.072	0.415	0.518
Plasma PC1	0.768	0.355	0.770	0.754	0.768
Plasma PC2	0.839	0.642	0.454	0.799	0.823

*Confounders included were sex, Rome III 9-15 sum score, and BMI

B. With interaction (model: community dissimilarity ~ Group * Variable)

Variable	Continuous variable			Binary variable		
	p (PD/control)	p (Variable)	p (Interaction)	p (PD/control)	p (Variable)	p (Interaction)
Total SCFAs	0.000	0.000	0.295	0.000	0.000	0.448
Acetic acid	0.000	0.000	0.380	0.000	0.002	0.775
Propionic acid	0.000	0.000	0.801	0.000	0.002	0.186
Butyric acid	0.000	0.000	0.144	0.000	0.000	0.040
Isobutyric acid	0.001	0.430	0.428	0.001	0.164	0.414
Isovaleric acid binary	0.000	0.098	0.272	0.000	0.098	0.272
Valeric acid binary	0.000	0.098	0.272	0.000	0.098	0.229
Stool NGAL	0.001	0.007	0.160	0.001	0.000	0.062
Stool calprotectin	0.000	0.105	0.734	0.000	0.038	0.702
Stool zonulin	0.000	0.000	0.144	0.000	0.001	0.038
Stool CXCL8	0.000	0.153	0.626	0.000	0.477	0.572
Stool IL1b	0.000	0.331	0.499	0.000	0.560	0.706
Stool IL4	0.000	0.682	0.158	0.000	0.168	0.773
Stool IL2	0.000	0.030	0.706	0.000	0.213	0.581
Stool PC	0.000	0.022	0.809	0.000	0.221	0.874
Plasma LBP	0.000	0.980	0.715	0.000	0.763	0.471
Plasma IFNg	0.000	0.896	0.823	0.000	0.047	0.763
Plasma CXCL8	0.000	0.678	0.707	0.000	0.956	0.589
Plasma IL6	0.000	0.409	0.035	0.000	0.747	0.177
Plasma PC1	0.000	0.753	0.317	0.000	0.960	0.498
Plasma PC2	0.000	0.797	0.260	0.000	0.487	0.125

Table S4. Differences in inflammatory and permeability marker/SCFA levels between enterotypes

Variable	Full data		Control-only		PD-only	
	p-value	adjusted p-value	p-value	adjusted p-value	p-value	adjusted p-value
Total SCFAs	0.1570	0.3660	0.5400	0.7700	0.1540	0.3770
Acetic acid	0.2112	0.4225	0.4969	0.7703	0.2617	0.5234
Propionic acid	<u>0.0443</u>	0.1361	0.3475	0.7703	0.1614	0.4842
Butyric acid	<u>0.0454</u>	0.1361	0.4226	0.7703	<u>0.0132</u>	0.0793
Isobutyric acid	0.9828	0.9828	0.6344	0.7703	0.3695	0.5542
Isovaleric acid binary	0.9387	0.9828	0.7703	0.7703	0.8600	0.8600
Valeric acid binary	0.9387	0.9828	0.7703	0.7703	0.8600	0.8600
Stool NGAL	<u>0.0010</u>	<u>0.0083</u>	<u>0.0019</u>	<u>0.0076</u>	0.4192	0.9357
Stool calprotectin	0.9692	0.9692	0.8117	0.8589	0.7018	0.9357
Stool zonulin	0.1142	0.3757	<u>0.0004</u>	<u>0.0029</u>	0.6335	0.9357
Stool CXCL8	0.2665	0.4264	0.1532	0.3065	0.9750	0.9750
Stool IL1b	0.1596	0.3757	0.1038	0.2767	0.9038	0.9750
Stool IL4	0.4579	0.5233	0.8589	0.8589	0.4514	0.9357
Stool IL2	0.3305	0.4407	0.4607	0.6143	0.4788	0.9357
Stool PC	0.1879	0.3757	0.3019	0.4831	0.3346	0.9357
Plasma LBP	0.9464	0.9464	0.7888	0.8233	0.6232	0.7479
Plasma IFNg	0.1233	0.4795	0.5424	0.8136	0.1341	0.7479
Plasma CXCL8	0.5156	0.9464	0.3773	0.8136	0.4267	0.7479
Plasma IL6	0.8950	0.9464	0.8233	0.8233	0.7523	0.7523
Plasma PC1	0.9179	0.9464	0.4506	0.8136	0.5412	0.7479
Plasma PC2	0.1598	0.4795	0.3261	0.8136	0.6222	0.7479

Adjusted p-value: multiple comparison corrected with Benjamini & Hochberg false discovery rate.

Table S5. Differentially abundant taxa for variables of interest.

A. SCFAs

SCFA	Taxon	Subset	Level	base Mean	log2 fold change	log2 fold change SE	stat	p-value	adjusted p-value	
Acetic acid	Bifidobacteriaceae	full data	Family	3205.45	-0.00021	0.00011	-1.88	0.0607	0.1647	
Acetic acid	Bifidobacteriaceae	full data, PD/C + confounder corrected	Family	3205.45	0.00003	0.00010	0.26	0.7969	0.9497	
Acetic acid	Bifidobacteriaceae	control only	Family	1414.94	-0.00001	0.00015	-0.05	0.9619	0.9619	
Acetic acid	Bifidobacteriaceae	PD only	Family	5127.66	-0.00021	0.00015	-1.44	0.1498	0.2588	
Acetic acid	Coriobacteriaceae	full data	Family	145.77	-0.00018	0.00005	-3.36	0.0008	0.0049	**
Acetic acid	Coriobacteriaceae	full data, PD/C + confounder corrected	Family	145.77	-0.00016	0.00006	-2.77	0.0056	0.0355	*
Acetic acid	Coriobacteriaceae	control only	Family	134.02	-0.00017	0.00008	-2.23	0.0258	0.1499	
Acetic acid	Coriobacteriaceae	PD only	Family	158.09	-0.00014	0.00008	-1.82	0.0695	0.2304	
Acetic acid	Enterobacteriaceae	full data	Family	624.32	0.00008	0.00015	0.53	0.5945	0.7040	
Acetic acid	Enterobacteriaceae	full data, PD/C + confounder corrected	Family	624.32	-0.00018	0.00057	-0.31	0.7530	0.9497	
Acetic acid	Enterobacteriaceae	control only	Family	534.86	-0.00032	0.00086	-0.37	0.7121	0.7958	
Acetic acid	Enterobacteriaceae	PD only	Family	692.93	-0.00029	0.00020	-1.46	0.1434	0.2588	
Acetic acid	Erysipelotrichaceae	full data	Family	193.88	0.00012	0.00006	1.91	0.0567	0.1647	
Acetic acid	Erysipelotrichaceae	full data, PD/C + confounder corrected	Family	193.88	0.00011	0.00007	1.72	0.0857	0.2715	
Acetic acid	Erysipelotrichaceae	control only	Family	182.70	-0.00001	0.00009	-0.13	0.8967	0.9465	
Acetic acid	Erysipelotrichaceae	PD only	Family	212.91	0.00029	0.00009	3.19	0.0014	0.0274	*
Acetic acid	Lachnospiraceae	full data	Family	14634.74	0.00013	0.00005	2.83	0.0047	0.0222	*
Acetic acid	Lachnospiraceae	full data, PD/C + confounder corrected	Family	14634.74	0.00012	0.00005	2.58	0.0098	0.0467	*
Acetic acid	Lachnospiraceae	control only	Family	16143.26	0.00009	0.00005	1.73	0.0836	0.3176	
Acetic acid	Lachnospiraceae	PD only	Family	13553.22	0.00017	0.00008	2.07	0.0389	0.1848	
Acetic acid	Peptostreptococcaceae	full data	Family	375.23	0.00048	0.00012	4.01	0.0001	0.0006	***
Acetic acid	Peptostreptococcaceae	full data, PD/C + confounder corrected	Family	375.23	0.00045	0.00012	3.67	0.0002	0.0023	**
Acetic acid	Peptostreptococcaceae	control only	Family	429.87	0.00048	0.00017	2.77	0.0056	0.0528	.
Acetic acid	Peptostreptococcaceae	PD only	Family	331.35	0.00046	0.00017	2.67	0.0077	0.0731	.
Acetic acid	Verrucomicrobiaceae	full data	Family	698.86	-0.00060	0.00014	-4.13	0.0000	0.0006	***
Acetic acid	Verrucomicrobiaceae	full data, PD/C + confounder corrected	Family	698.86	-0.00057	0.00015	-3.76	0.0002	0.0023	**
Acetic acid	Verrucomicrobiaceae	control only	Family	702.56	-0.00075	0.00020	-3.70	0.0002	0.0042	**
Acetic acid	Verrucomicrobiaceae	PD only	Family	705.09	-0.00051	0.00021	-2.37	0.0177	0.1122	
Acetic acid	Akkermansia	full data	Genus	833.65	-0.00068	0.00015	-4.63	0.0000	0.0000	***
Acetic acid	Akkermansia	full data, PD/C + confounder corrected	Genus	833.65	-0.00066	0.00015	-4.36	0.0000	0.0001	***
Acetic acid	Akkermansia	control only	Genus	681.42	-0.00069	0.00019	-3.59	0.0003	0.0028	**
Acetic acid	Akkermansia	PD only	Genus	953.40	-0.00066	0.00022	-2.98	0.0029	0.0308	*
Acetic acid	Alistipes	full data	Genus	1776.39	-0.00011	0.00007	-1.66	0.0978	0.2103	
Acetic acid	Alistipes	full data, PD/C + confounder corrected	Genus	1776.39	-0.00008	0.00007	-1.21	0.2245	0.4157	
Acetic acid	Alistipes	control only	Genus	1616.26	-0.00005	0.00009	-0.56	0.5747	0.6920	
Acetic acid	Alistipes	PD only	Genus	1939.52	-0.00015	0.00010	-1.55	0.1204	0.5055	
Acetic acid	Anaerotruncus	full data	Genus	73.92	-0.00040	0.00010	-3.85	0.0001	0.0010	**
Acetic acid	Anaerotruncus	full data, PD/C + confounder corrected	Genus	73.92	-0.00035	0.00011	-3.30	0.0010	0.0060	**
Acetic acid	Anaerotruncus	control only	Genus	81.77	-0.00062	0.00015	-4.20	0.0000	0.0004	***
Acetic acid	Anaerotruncus	PD only	Genus	60.74	-0.00013	0.00014	-0.93	0.3520	0.6987	
Acetic acid	Bacteroides	full data	Genus	2746.72	0.00007	0.00007	1.08	0.2801	0.4302	
Acetic acid	Bacteroides	full data, PD/C + confounder corrected	Genus	2746.72	0.00009	0.00007	1.27	0.2041	0.4157	
Acetic acid	Bacteroides	control only	Genus	2717.10	0.00026	0.00010	2.60	0.0093	0.0399	*
Acetic acid	Bacteroides	PD only	Genus	2862.35	-0.00013	0.00010	-1.33	0.1830	0.5055	
Acetic acid	Bifidobacterium	full data	Genus	3383.49	-0.00020	0.00011	-1.80	0.0722	0.2013	
Acetic acid	Bifidobacterium	full data, PD/C + confounder corrected	Genus	3383.49	0.00005	0.00010	0.50	0.6163	0.7794	
Acetic acid	Bifidobacterium	control only	Genus	1334.56	0.00005	0.00015	0.34	0.7323	0.8074	
Acetic acid	Bifidobacterium	PD only	Genus	5469.37	-0.00024	0.00015	-1.56	0.1176	0.5055	
Acetic acid	Butyrivibrio	full data	Genus	201.42	0.00033	0.00006	5.43	0.0000	0.0000	***
Acetic acid	Butyrivibrio	full data, PD/C + confounder corrected	Genus	201.42	0.00028	0.00006	4.59	0.0000	0.0001	***
Acetic acid	Butyrivibrio	control only	Genus	270.47	0.00032	0.00009	3.43	0.0006	0.0044	**
Acetic acid	Butyrivibrio	PD only	Genus	138.83	0.00021	0.00008	2.66	0.0079	0.0681	.
Acetic acid	Clostridium sensu stricto	full data	Genus	360.30	0.00052	0.00014	3.60	0.0003	0.0022	**
Acetic acid	Clostridium sensu stricto	full data, PD/C + confounder corrected	Genus	360.30	0.00053	0.00015	3.56	0.0004	0.0026	**
Acetic acid	Clostridium sensu stricto	control only	Genus	398.65	0.00033	0.00018	1.83	0.0670	0.2058	
Acetic acid	Clostridium sensu stricto	PD only	Genus	313.22	0.00086	0.00022	3.85	0.0001	0.0026	**
Acetic acid	Escherichia/Shigella	full data	Genus	413.05	-0.00041	0.00015	-2.71	0.0068	0.0363	*
Acetic acid	Escherichia/Shigella	full data, PD/C + confounder corrected	Genus	413.05	-0.00045	0.00016	-2.91	0.0036	0.0193	*
Acetic acid	Escherichia/Shigella	control only	Genus	285.22	-0.00088	0.00022	-4.02	0.0001	0.0006	***

Acetic acid	Escherichia/Shigella	PD only	Genus	529.75	-0.00029	0.00021	-1.41	0.1583	0.5055	
Acetic acid	Flavonifractor	full data	Genus	212.39	-0.00036	0.00011	-3.43	0.0006	0.0037	**
Acetic acid	Flavonifractor	full data, PD/C + confounder corrected	Genus	212.39	-0.00041	0.00011	-3.90	0.0001	0.0008	***
Acetic acid	Flavonifractor	control only	Genus	181.49	-0.00024	0.00015	-1.58	0.1150	0.3091	
Acetic acid	Flavonifractor	PD only	Genus	239.89	-0.00054	0.00015	-3.68	0.0002	0.0033	**
Acetic acid	Intestinimonas	full data	Genus	95.18	-0.00008	0.00009	-0.83	0.4058	0.5133	
Acetic acid	Intestinimonas	full data, PD/C + confounder corrected	Genus	95.18	-0.00011	0.00009	-1.17	0.2417	0.4157	
Acetic acid	Intestinimonas	control only	Genus	73.57	-0.00036	0.00013	-2.79	0.0053	0.0254	*
Acetic acid	Intestinimonas	PD only	Genus	111.58	0.00010	0.00013	0.82	0.4105	0.7275	
Acetic acid	Phascolarctobacterium	full data	Genus	545.74	-0.00048	0.00019	-2.47	0.0135	0.0644	.
Acetic acid	Phascolarctobacterium	full data, PD/C + confounder corrected	Genus	545.74	-0.00037	0.00086	-0.43	0.6677	0.7975	
Acetic acid	Phascolarctobacterium	control only	Genus	506.82	-0.00121	0.00024	-5.02	0.0000	0.0000	***
Acetic acid	Phascolarctobacterium	PD only	Genus	578.80	-0.00008	0.00087	-0.10	0.9241	0.9705	
Acetic acid	Romboutsia	full data	Genus	144.79	0.00076	0.00013	5.95	0.0000	0.0000	***
Acetic acid	Romboutsia	full data, PD/C + confounder corrected	Genus	144.79	0.00068	0.00013	5.33	0.0000	0.0000	***
Acetic acid	Romboutsia	control only	Genus	180.01	0.00057	0.00019	3.03	0.0024	0.0149	*
Acetic acid	Romboutsia	PD only	Genus	115.59	0.00072	0.00016	4.50	0.0000	0.0003	***
Acetic acid	Roseburia	full data	Genus	3994.13	0.00045	0.00009	5.02	0.0000	0.0000	***
Acetic acid	Roseburia	full data, PD/C + confounder corrected	Genus	3994.13	0.00042	0.00009	4.51	0.0000	0.0001	***
Acetic acid	Roseburia	control only	Genus	5167.52	0.00046	0.00010	4.40	0.0000	0.0002	***
Acetic acid	Roseburia	PD only	Genus	2885.18	0.00038	0.00015	2.53	0.0114	0.0814	.
Acetic acid	Sporobacter	full data	Genus	9.45	-0.00015	0.00009	-1.80	0.0723	0.2013	
Acetic acid	Sporobacter	full data, PD/C + confounder corrected	Genus	9.45	-0.00014	0.00009	-1.54	0.1240	0.3251	
Acetic acid	Sporobacter	control only	Genus	10.56	-0.00037	0.00013	-2.91	0.0036	0.0193	*
Acetic acid	Sporobacter	PD only	Genus	8.46	0.00001	0.00012	0.06	0.9548	0.9705	
Butyric acid	Bifidobacteriaceae	full data	Family	3205.45	-0.00121	0.00033	-3.67	0.0002	0.0016	**
Butyric acid	Bifidobacteriaceae	full data, PD/C + confounder corrected	Family	3205.45	-0.00029	0.00033	-0.89	0.3726	0.6323	
Butyric acid	Bifidobacteriaceae	control only	Family	1414.94	-0.00023	0.00041	-0.56	0.5767	0.6088	
Butyric acid	Bifidobacteriaceae	PD only	Family	5127.66	-0.00185	0.00058	-3.20	0.0014	0.0132	*
Butyric acid	Coriobacteriaceae	full data	Family	145.77	-0.00047	0.00017	-2.83	0.0047	0.0179	*
Butyric acid	Coriobacteriaceae	full data, PD/C + confounder corrected	Family	145.77	-0.00039	0.00018	-2.19	0.0287	0.1091	
Butyric acid	Coriobacteriaceae	control only	Family	134.02	-0.00024	0.00022	-1.12	0.2632	0.3572	
Butyric acid	Coriobacteriaceae	PD only	Family	158.09	-0.00070	0.00033	-2.15	0.0313	0.0992	.
Butyric acid	Enterobacteriaceae	full data	Family	624.32	-0.00124	0.00046	-2.68	0.0074	0.0234	*
Butyric acid	Enterobacteriaceae	full data, PD/C + confounder corrected	Family	624.32	-0.00124	0.00049	-2.52	0.0118	0.0560	.
Butyric acid	Enterobacteriaceae	control only	Family	534.86	-0.00269	0.00061	-4.39	0.0000	0.0002	***
Butyric acid	Enterobacteriaceae	PD only	Family	692.93	-0.00081	0.00081	-1.00	0.3150	0.4987	
Butyric acid	Erysipelotrichaceae	full data	Family	193.88	-0.00012	0.00019	-0.64	0.5244	0.7251	
Butyric acid	Erysipelotrichaceae	full data, PD/C + confounder corrected	Family	193.88	-0.00011	0.00021	-0.51	0.6068	0.7206	
Butyric acid	Erysipelotrichaceae	control only	Family	182.70	-0.00031	0.00023	-1.32	0.1853	0.2804	
Butyric acid	Erysipelotrichaceae	PD only	Family	212.91	0.00088	0.00040	2.18	0.0291	0.0992	.
Butyric acid	Lachnospiraceae	full data	Family	14634.74	0.00048	0.00014	3.50	0.0005	0.0022	**
Butyric acid	Lachnospiraceae	full data, PD/C + confounder corrected	Family	14634.74	0.00045	0.00014	3.13	0.0018	0.0112	*
Butyric acid	Lachnospiraceae	control only	Family	16143.26	0.00028	0.00014	2.03	0.0419	0.1593	
Butyric acid	Lachnospiraceae	PD only	Family	13553.22	0.00110	0.00032	3.41	0.0006	0.0122	*
Butyric acid	Peptostreptococcaceae	full data	Family	375.23	0.00158	0.00036	4.38	0.0000	0.0002	***
Butyric acid	Peptostreptococcaceae	full data, PD/C + confounder corrected	Family	375.23	0.00156	0.00038	4.08	0.0000	0.0008	***
Butyric acid	Peptostreptococcaceae	control only	Family	429.87	0.00152	0.00043	3.51	0.0004	0.0028	**
Butyric acid	Peptostreptococcaceae	PD only	Family	331.35	0.00174	0.00073	2.38	0.0173	0.0820	.
Butyric acid	Verrucomicrobiaceae	full data	Family	698.86	-0.00171	0.00045	-3.84	0.0001	0.0012	**
Butyric acid	Verrucomicrobiaceae	full data, PD/C + confounder corrected	Family	698.86	-0.00168	0.00048	-3.53	0.0004	0.0040	**
Butyric acid	Verrucomicrobiaceae	control only	Family	702.56	-0.00206	0.00054	-3.82	0.0001	0.0013	**
Butyric acid	Verrucomicrobiaceae	PD only	Family	705.09	-0.00136	0.00089	-1.53	0.1261	0.2395	
Butyric acid	Akkermansia	full data	Genus	833.65	-0.00212	0.00045	-4.71	0.0000	0.0000	***
Butyric acid	Akkermansia	full data, PD/C + confounder corrected	Genus	833.65	-0.00204	0.00048	-4.21	0.0000	0.0003	***
Butyric acid	Akkermansia	control only	Genus	681.42	-0.00199	0.00052	-3.83	0.0001	0.0014	**
Butyric acid	Akkermansia	PD only	Genus	953.40	-0.00206	0.00091	-2.26	0.0240	0.1722	
Butyric acid	Alistipes	full data	Genus	1776.39	-0.00052	0.00020	-2.61	0.0090	0.0379	*
Butyric acid	Alistipes	full data, PD/C + confounder corrected	Genus	1776.39	-0.00040	0.00021	-1.89	0.0593	0.2317	
Butyric acid	Alistipes	control only	Genus	1616.26	-0.00032	0.00026	-1.25	0.2121	0.4515	
Butyric acid	Alistipes	PD only	Genus	1939.52	-0.00065	0.00040	-1.63	0.1021	0.3991	
Butyric acid	Anaerotruncus	full data	Genus	73.92	-0.00138	0.00032	-4.36	0.0000	0.0001	***
Butyric acid	Anaerotruncus	full data, PD/C + confounder corrected	Genus	73.92	-0.00141	0.00033	-4.21	0.0000	0.0003	***
Butyric acid	Anaerotruncus	control only	Genus	81.77	-0.00136	0.00041	-3.36	0.0008	0.0048	**
Butyric acid	Anaerotruncus	PD only	Genus	60.74	-0.00169	0.00057	-2.97	0.0030	0.0322	*
Butyric acid	Bacteroides	full data	Genus	2746.72	0.00017	0.00021	0.82	0.4130	0.5920	

Butyric acid	Bacteroides	full data, PD/C + confounder corrected	Genus	2746.72	0.00019	0.00022	0.90	0.3707	0.6641	
Butyric acid	Bacteroides	control only	Genus	2717.10	0.00053	0.00028	1.94	0.0530	0.2111	
Butyric acid	Bacteroides	PD only	Genus	2862.35	-0.00030	0.00041	-0.73	0.4644	0.7132	
Butyric acid	Bifidobacterium	full data	Genus	3383.49	-0.00122	0.00033	-3.64	0.0003	0.0023	**
Butyric acid	Bifidobacterium	full data, PD/C + confounder corrected	Genus	3383.49	-0.00022	0.00033	-0.69	0.4917	0.6740	
Butyric acid	Bifidobacterium	control only	Genus	1334.56	-0.00007	0.00039	-0.18	0.8536	0.8953	
Butyric acid	Bifidobacterium	PD only	Genus	5469.37	-0.00196	0.00059	-3.33	0.0009	0.0123	*
Butyric acid	Butyricoccus	full data	Genus	201.42	0.00071	0.00020	3.58	0.0003	0.0025	**
Butyric acid	Butyricoccus	full data, PD/C + confounder corrected	Genus	201.42	0.00046	0.00020	2.30	0.0212	0.1014	
Butyric acid	Butyricoccus	control only	Genus	270.47	0.00045	0.00027	1.67	0.0950	0.2918	
Butyric acid	Butyricoccus	PD only	Genus	138.83	0.00066	0.00034	1.91	0.0558	0.2665	
Butyric acid	Clostridium sensu stricto	full data	Genus	360.30	0.00061	0.00045	1.37	0.1694	0.3407	
Butyric acid	Clostridium sensu stricto	full data, PD/C + confounder corrected	Genus	360.30	0.00081	0.00048	1.69	0.0907	0.3051	
Butyric acid	Clostridium sensu stricto	control only	Genus	398.65	0.00094	0.00049	1.93	0.0540	0.2111	
Butyric acid	Clostridium sensu stricto	PD only	Genus	313.22	0.00043	0.00094	0.45	0.6508	0.9080	
Butyric acid	Escherichia/Shigella	full data	Genus	413.05	-0.00158	0.00047	-3.39	0.0007	0.0043	**
Butyric acid	Escherichia/Shigella	full data, PD/C + confounder corrected	Genus	413.05	-0.00151	0.00050	-3.04	0.0024	0.0171	*
Butyric acid	Escherichia/Shigella	control only	Genus	285.22	-0.00236	0.00060	-3.90	0.0001	0.0014	**
Butyric acid	Escherichia/Shigella	PD only	Genus	529.75	-0.00103	0.00086	-1.21	0.2280	0.5381	
Butyric acid	Flavonifractor	full data	Genus	212.39	-0.00085	0.00033	-2.59	0.0097	0.0379	*
Butyric acid	Flavonifractor	full data, PD/C + confounder corrected	Genus	212.39	-0.00079	0.00034	-2.31	0.0208	0.1014	
Butyric acid	Flavonifractor	control only	Genus	181.49	-0.00047	0.00041	-1.14	0.2559	0.4586	
Butyric acid	Flavonifractor	PD only	Genus	239.89	-0.00176	0.00062	-2.83	0.0046	0.0398	*
Butyric acid	Intestinimonas	full data	Genus	95.18	-0.00088	0.00028	-3.16	0.0016	0.0084	**
Butyric acid	Intestinimonas	full data, PD/C + confounder corrected	Genus	95.18	-0.00089	0.00028	-3.17	0.0015	0.0130	*
Butyric acid	Intestinimonas	control only	Genus	73.57	-0.00088	0.00035	-2.51	0.0121	0.0649	.
Butyric acid	Intestinimonas	PD only	Genus	111.58	-0.00052	0.00052	-1.01	0.3120	0.5834	
Butyric acid	Phascolarctobacterium	full data	Genus	545.74	-0.00118	0.00250	-0.47	0.6359	0.7192	
Butyric acid	Phascolarctobacterium	full data, PD/C + confounder corrected	Genus	545.74	-0.00112	0.00275	-0.41	0.6843	0.8173	
Butyric acid	Phascolarctobacterium	control only	Genus	506.82	-0.00282	0.00071	-3.95	0.0001	0.0014	**
Butyric acid	Phascolarctobacterium	PD only	Genus	578.80	-0.00004	0.00359	-0.01	0.9909	0.9976	
Butyric acid	Romboutsia	full data	Genus	144.79	0.00271	0.00035	7.69	0.0000	0.0000	***
Butyric acid	Romboutsia	full data, PD/C + confounder corrected	Genus	144.79	0.00263	0.00038	6.96	0.0000	0.0000	***
Butyric acid	Romboutsia	control only	Genus	180.01	0.00242	0.00045	5.40	0.0000	0.0000	***
Butyric acid	Romboutsia	PD only	Genus	115.59	0.00279	0.00065	4.27	0.0000	0.0004	***
Butyric acid	Roseburia	full data	Genus	3994.13	0.00165	0.00027	6.03	0.0000	0.0000	***
Butyric acid	Roseburia	full data, PD/C + confounder corrected	Genus	3994.13	0.00160	0.00029	5.56	0.0000	0.0000	***
Butyric acid	Roseburia	control only	Genus	5167.52	0.00102	0.00029	3.51	0.0005	0.0039	**
Butyric acid	Roseburia	PD only	Genus	2885.18	0.00296	0.00057	5.20	0.0000	0.0000	***
Butyric acid	Sporobacter	full data	Genus	9.45	-0.00072	0.00027	-2.67	0.0076	0.0365	*
Butyric acid	Sporobacter	full data, PD/C + confounder corrected	Genus	9.45	-0.00068	0.00028	-2.41	0.0158	0.0969	.
Butyric acid	Sporobacter	control only	Genus	10.56	-0.00126	0.00036	-3.45	0.0006	0.0040	**
Butyric acid	Sporobacter	PD only	Genus	8.46	-0.00007	0.00048	-0.14	0.8924	0.9801	
Propionic acid	Bifidobacteriaceae	full data	Family	3205.45	-0.00111	0.00042	-2.65	0.0080	0.0393	*
Propionic acid	Bifidobacteriaceae	full data, PD/C + confounder corrected	Family	3205.45	-0.00005	0.00040	-0.11	0.9101	0.9594	
Propionic acid	Bifidobacteriaceae	control only	Family	1414.94	-0.00020	0.00051	-0.38	0.7006	0.8319	
Propionic acid	Bifidobacteriaceae	PD only	Family	5127.66	-0.00132	0.00069	-1.90	0.0578	0.2196	
Propionic acid	Coriobacteriaceae	full data	Family	145.77	-0.00054	0.00020	-2.64	0.0083	0.0393	*
Propionic acid	Coriobacteriaceae	full data, PD/C + confounder corrected	Family	145.77	-0.00043	0.00022	-2.01	0.0440	0.2787	
Propionic acid	Coriobacteriaceae	control only	Family	134.02	-0.00028	0.00027	-1.02	0.3078	0.4499	
Propionic acid	Coriobacteriaceae	PD only	Family	158.09	-0.00087	0.00037	-2.36	0.0183	0.1160	
Propionic acid	Enterobacteriaceae	full data	Family	624.32	0.00062	0.00058	1.08	0.2803	0.4815	
Propionic acid	Enterobacteriaceae	full data, PD/C + confounder corrected	Family	624.32	-0.00082	0.00220	-0.37	0.7085	0.9594	
Propionic acid	Enterobacteriaceae	control only	Family	534.86	0.00155	0.00076	2.04	0.0411	0.2741	
Propionic acid	Enterobacteriaceae	PD only	Family	692.93	-0.00007	0.00094	-0.08	0.9394	0.9916	
Propionic acid	Erysipelotrichaceae	full data	Family	193.88	0.00034	0.00024	1.43	0.1520	0.3375	
Propionic acid	Erysipelotrichaceae	full data, PD/C + confounder corrected	Family	193.88	0.00035	0.00025	1.42	0.1554	0.4218	
Propionic acid	Erysipelotrichaceae	control only	Family	182.70	-0.00004	0.00029	-0.13	0.8941	0.9161	
Propionic acid	Erysipelotrichaceae	PD only	Family	212.91	0.00121	0.00045	2.70	0.0070	0.0903	.
Propionic acid	Lachnospiraceae	full data	Family	14634.74	0.00047	0.00017	2.72	0.0066	0.0393	*
Propionic acid	Lachnospiraceae	full data, PD/C + confounder corrected	Family	14634.74	0.00044	0.00018	2.42	0.0156	0.1480	
Propionic acid	Lachnospiraceae	control only	Family	16143.26	0.00031	0.00017	1.85	0.0645	0.2741	
Propionic acid	Lachnospiraceae	PD only	Family	13553.22	0.00076	0.00039	1.97	0.0488	0.2196	
Propionic acid	Peptostreptococcaceae	full data	Family	375.23	0.00147	0.00047	3.17	0.0015	0.0292	*
Propionic acid	Peptostreptococcaceae	full data, PD/C + confounder corrected	Family	375.23	0.00144	0.00049	2.94	0.0032	0.0617	.

Propionic acid	Peptostreptococcaceae	control only	Family	429.87	0.00101	0.00056	1.80	0.0721	0.2741	
Propionic acid	Peptostreptococcaceae	PD only	Family	331.35	0.00227	0.00088	2.59	0.0095	0.0903	.
Propionic acid	Verrucomicrobiaceae	full data	Family	698.86	-0.00001	0.00056	-0.02	0.9866	0.9978	
Propionic acid	Verrucomicrobiaceae	full data, PD/C + confounder corrected	Family	698.86	-0.00011	0.00059	-0.18	0.8561	0.9594	
Propionic acid	Verrucomicrobiaceae	control only	Family	702.56	-0.00007	0.00070	-0.11	0.9161	0.9161	
Propionic acid	Verrucomicrobiaceae	PD only	Family	705.09	-0.00023	0.00104	-0.22	0.8250	0.9845	
Propionic acid	Akkermansia	full data	Genus	833.65	-0.00108	0.00057	-1.88	0.0607	0.2088	
Propionic acid	Akkermansia	full data, PD/C + confounder corrected	Genus	833.65	-0.00090	0.00061	-1.49	0.1368	0.3269	
Propionic acid	Akkermansia	control only	Genus	681.42	-0.00064	0.00067	-0.95	0.3414	0.5656	
Propionic acid	Akkermansia	PD only	Genus	953.40	-0.00138	0.00108	-1.28	0.1991	0.5709	
Propionic acid	Alistipes	full data	Genus	1776.39	-0.00050	0.00025	-2.03	0.0424	0.1821	
Propionic acid	Alistipes	full data, PD/C + confounder corrected	Genus	1776.39	-0.00037	0.00026	-1.43	0.1517	0.3433	
Propionic acid	Alistipes	control only	Genus	1616.26	-0.00030	0.00031	-0.97	0.3303	0.5656	
Propionic acid	Alistipes	PD only	Genus	1939.52	-0.00071	0.00045	-1.57	0.1174	0.3882	
Propionic acid	Anaerotruncus	full data	Genus	73.92	-0.00143	0.00040	-3.61	0.0003	0.0032	**
Propionic acid	Anaerotruncus	full data, PD/C + confounder corrected	Genus	73.92	-0.00138	0.00041	-3.33	0.0009	0.0094	**
Propionic acid	Anaerotruncus	control only	Genus	81.77	-0.00166	0.00051	-3.28	0.0011	0.0131	*
Propionic acid	Anaerotruncus	PD only	Genus	60.74	-0.00100	0.00067	-1.51	0.1309	0.4021	
Propionic acid	Bacteroides	full data	Genus	2746.72	0.00042	0.00026	1.61	0.1072	0.2304	
Propionic acid	Bacteroides	full data, PD/C + confounder corrected	Genus	2746.72	0.00041	0.00026	1.54	0.1232	0.3269	
Propionic acid	Bacteroides	control only	Genus	2717.10	0.00080	0.00033	2.41	0.0161	0.0884	.
Propionic acid	Bacteroides	PD only	Genus	2862.35	0.00016	0.00047	0.33	0.7409	0.8762	
Propionic acid	Bifidobacterium	full data	Genus	3383.49	-0.00102	0.00042	-2.42	0.0155	0.0955	.
Propionic acid	Bifidobacterium	full data, PD/C + confounder corrected	Genus	3383.49	0.00015	0.00040	0.36	0.7154	0.9361	
Propionic acid	Bifidobacterium	control only	Genus	1334.56	0.00002	0.00049	0.04	0.9712	0.9911	
Propionic acid	Bifidobacterium	PD only	Genus	5469.37	-0.00133	0.00071	-1.86	0.0623	0.2677	
Propionic acid	Butyricicoccus	full data	Genus	201.42	0.00123	0.00023	5.23	0.0000	0.0000	***
Propionic acid	Butyricicoccus	full data, PD/C + confounder corrected	Genus	201.42	0.00099	0.00024	4.14	0.0000	0.0005	***
Propionic acid	Butyricicoccus	control only	Genus	270.47	0.00104	0.00032	3.23	0.0012	0.0131	*
Propionic acid	Butyricicoccus	PD only	Genus	138.83	0.00111	0.00037	2.99	0.0028	0.0406	*
Propionic acid	Clostridium sensu stricto	full data	Genus	360.30	0.00087	0.00055	1.57	0.1168	0.2392	
Propionic acid	Clostridium sensu stricto	full data, PD/C + confounder corrected	Genus	360.30	0.00112	0.00058	1.93	0.0532	0.2285	
Propionic acid	Clostridium sensu stricto	control only	Genus	398.65	0.00056	0.00061	0.91	0.3633	0.5785	
Propionic acid	Clostridium sensu stricto	PD only	Genus	313.22	0.00265	0.00109	2.44	0.0148	0.1511	
Propionic acid	Escherichia/Shigella	full data	Genus	413.05	-0.00099	0.00058	-1.68	0.0920	0.2199	
Propionic acid	Escherichia/Shigella	full data, PD/C + confounder corrected	Genus	413.05	-0.00092	0.00061	-1.50	0.1334	0.3269	
Propionic acid	Escherichia/Shigella	control only	Genus	285.22	-0.00025	0.00076	-0.33	0.7407	0.9329	
Propionic acid	Escherichia/Shigella	PD only	Genus	529.75	-0.00086	0.00099	-0.87	0.3824	0.7148	
Propionic acid	Flavonifractor	full data	Genus	212.39	-0.00081	0.00041	-1.99	0.0468	0.1829	
Propionic acid	Flavonifractor	full data, PD/C + confounder corrected	Genus	212.39	-0.00080	0.00042	-1.89	0.0587	0.2285	
Propionic acid	Flavonifractor	control only	Genus	181.49	-0.00028	0.00052	-0.54	0.5861	0.7637	
Propionic acid	Flavonifractor	PD only	Genus	239.89	-0.00155	0.00072	-2.16	0.0307	0.1942	
Propionic acid	Intestinimonas	full data	Genus	95.18	-0.00033	0.00035	-0.93	0.3498	0.5187	
Propionic acid	Intestinimonas	full data, PD/C + confounder corrected	Genus	95.18	-0.00039	0.00035	-1.11	0.2659	0.5198	
Propionic acid	Intestinimonas	control only	Genus	73.57	-0.00031	0.00044	-0.69	0.4886	0.7305	
Propionic acid	Intestinimonas	PD only	Genus	111.58	0.00029	0.00060	0.48	0.6316	0.8761	
Propionic acid	Phascolarctobacterium	full data	Genus	545.74	-0.00115	0.00320	-0.36	0.7208	0.8157	
Propionic acid	Phascolarctobacterium	full data, PD/C + confounder corrected	Genus	545.74	-0.00093	0.00340	-0.27	0.7837	0.9361	
Propionic acid	Phascolarctobacterium	control only	Genus	506.82	-0.00230	0.00090	-2.55	0.0109	0.0884	.
Propionic acid	Phascolarctobacterium	PD only	Genus	578.80	-0.00015	0.00416	-0.04	0.9713	0.9713	
Propionic acid	Romboutsia	full data	Genus	144.79	0.00304	0.00049	6.26	0.0000	0.0000	***
Propionic acid	Romboutsia	full data, PD/C + confounder corrected	Genus	144.79	0.00305	0.00051	5.93	0.0000	0.0000	***
Propionic acid	Romboutsia	control only	Genus	180.01	0.00202	0.00060	3.38	0.0007	0.0131	*
Propionic acid	Romboutsia	PD only	Genus	115.59	0.00448	0.00085	5.29	0.0000	0.0000	***
Propionic acid	Roseburia	full data	Genus	3994.13	0.00171	0.00034	5.00	0.0000	0.0000	***
Propionic acid	Roseburia	full data, PD/C + confounder corrected	Genus	3994.13	0.00170	0.00036	4.79	0.0000	0.0000	***
Propionic acid	Roseburia	control only	Genus	5167.52	0.00137	0.00035	3.89	0.0001	0.0043	**
Propionic acid	Roseburia	PD only	Genus	2885.18	0.00215	0.00070	3.09	0.0020	0.0406	*
Propionic acid	Sporobacter	full data	Genus	9.45	-0.00070	0.00033	-2.13	0.0329	0.1570	
Propionic acid	Sporobacter	full data, PD/C + confounder corrected	Genus	9.45	-0.00066	0.00034	-1.96	0.0501	0.2285	
Propionic acid	Sporobacter	control only	Genus	10.56	-0.00091	0.00043	-2.09	0.0367	0.1434	
Propionic acid	Sporobacter	PD only	Genus	8.46	-0.00030	0.00057	-0.54	0.5903	0.8753	

B. Stool inflammatory markers

Marker	Taxon	Subset	Level	base Mean	log2 fold change	log2 fold change SE	stat	p-value	adjusted p-value	
Stool NGAL	Bacteroidaceae	full data	Family	2812.21	0.00657	0.00150	4.39	0.0000	0.0002	***
Stool NGAL	Bacteroidaceae	full data, PD/C + confounder corrected	Family	2812.21	0.00625	0.00147	4.24	0.0000	0.0004	***
Stool NGAL	Bacteroidaceae	control only	Family	2770.80	0.00986	0.00205	4.80	0.0000	0.0000	***
Stool NGAL	Bacteroidaceae	PD only	Family	2900.39	0.00449	0.00223	2.01	0.0441	0.1675	
Stool NGAL	Bifidobacteriaceae	full data	Family	3002.57	0.00846	0.00249	3.40	0.0007	0.0042	**
Stool NGAL	Bifidobacteriaceae	full data, PD/C + confounder corrected	Family	3002.57	0.00435	0.00228	1.91	0.0563	0.2140	
Stool NGAL	Bifidobacteriaceae	control only	Family	1434.57	-0.00032	0.00338	-0.09	0.9245	0.9758	
Stool NGAL	Bifidobacteriaceae	PD only	Family	4832.79	0.01291	0.00322	4.01	0.0001	0.0012	**
Stool NGAL	Prevotellaceae	full data	Family	1632.19	-0.01424	0.00389	-3.66	0.0003	0.0024	**
Stool NGAL	Prevotellaceae	full data, PD/C + confounder corrected	Family	1632.19	-0.01367	0.00388	-3.52	0.0004	0.0041	**
Stool NGAL	Prevotellaceae	control only	Family	2064.45	-0.01663	0.00491	-3.39	0.0007	0.0067	**
Stool NGAL	Prevotellaceae	PD only	Family	1311.17	-0.00238	0.01948	-0.12	0.9028	0.9403	
Stool NGAL	Streptococcaceae	full data	Family	174.08	0.00508	0.00209	2.44	0.0148	0.0562	.
Stool NGAL	Streptococcaceae	full data, PD/C + confounder corrected	Family	174.08	0.00570	0.00204	2.79	0.0053	0.0334	*
Stool NGAL	Streptococcaceae	control only	Family	185.38	0.00402	0.00302	1.33	0.1823	0.4330	
Stool NGAL	Streptococcaceae	PD only	Family	157.87	0.00829	0.00265	3.13	0.0018	0.0167	*
Stool NGAL	Verrucomicrobiaceae	full data	Family	727.72	-0.00291	0.00336	-0.87	0.3863	0.5621	
Stool NGAL	Verrucomicrobiaceae	full data, PD/C + confounder corrected	Family	727.72	-0.00677	0.01145	-0.59	0.5545	0.8104	
Stool NGAL	Verrucomicrobiaceae	control only	Family	714.21	0.00239	0.00458	0.52	0.6015	0.8791	
Stool NGAL	Verrucomicrobiaceae	PD only	Family	734.13	-0.01397	0.00479	-2.91	0.0036	0.0225	*
Stool NGAL	Victivallaceae	full data	Family	143.18	-0.00971	0.00379	-2.56	0.0104	0.0492	*
Stool NGAL	Victivallaceae	full data, PD/C + confounder corrected	Family	143.18	-0.01700	0.01103	-1.54	0.1234	0.3348	
Stool NGAL	Victivallaceae	control only	Family	170.99	-0.01086	0.00490	-2.22	0.0265	0.1681	
Stool NGAL	Victivallaceae	PD only	Family	104.30	-0.00462	0.00569	-0.81	0.4169	0.7888	
Stool NGAL	Akkermansia	full data	Genus	867.68	-0.00784	0.00340	-2.31	0.0211	0.1007	
Stool NGAL	Akkermansia	full data, PD/C + confounder corrected	Genus	867.68	-0.00808	0.00342	-2.36	0.0181	0.1109	
Stool NGAL	Akkermansia	control only	Genus	694.86	-0.00092	0.00450	-0.20	0.8381	0.9835	
Stool NGAL	Akkermansia	PD only	Genus	1008.54	-0.01569	0.00493	-3.19	0.0014	0.0207	*
Stool NGAL	Anaerotruncus	full data	Genus	75.84	-0.00610	0.00239	-2.55	0.0108	0.0682	.
Stool NGAL	Anaerotruncus	full data, PD/C + confounder corrected	Genus	75.84	-0.00737	0.00238	-3.10	0.0019	0.0279	*
Stool NGAL	Anaerotruncus	control only	Genus	83.81	-0.01000	0.00333	-3.00	0.0027	0.0294	*
Stool NGAL	Anaerotruncus	PD only	Genus	61.79	-0.00256	0.00328	-0.78	0.4365	0.8587	
Stool NGAL	Bacteroides	full data	Genus	2804.94	0.00543	0.00153	3.54	0.0004	0.0137	*
Stool NGAL	Bacteroides	full data, PD/C + confounder corrected	Genus	2804.94	0.00534	0.00145	3.68	0.0002	0.0102	*
Stool NGAL	Bacteroides	control only	Genus	2747.89	0.00863	0.00208	4.14	0.0000	0.0015	**
Stool NGAL	Bacteroides	PD only	Genus	2929.70	0.00337	0.00228	1.48	0.1383	0.5408	
Stool NGAL	Bifidobacterium	full data	Genus	3142.78	0.00855	0.00250	3.42	0.0006	0.0137	*
Stool NGAL	Bifidobacterium	full data, PD/C + confounder corrected	Genus	3142.78	0.00401	0.00230	1.74	0.0815	0.2697	
Stool NGAL	Bifidobacterium	control only	Genus	1355.14	-0.00037	0.00322	-0.11	0.9092	0.9835	
Stool NGAL	Bifidobacterium	PD only	Genus	5037.87	0.01225	0.00334	3.67	0.0002	0.0080	**
Stool NGAL	Coprococcus	full data	Genus	1657.12	-0.00050	0.00249	-0.20	0.8399	0.9933	
Stool NGAL	Coprococcus	full data, PD/C + confounder corrected	Genus	1657.12	-0.00238	0.00248	-0.96	0.3367	0.6374	
Stool NGAL	Coprococcus	control only	Genus	1613.54	-0.00771	0.00306	-2.52	0.0118	0.0736	.
Stool NGAL	Coprococcus	PD only	Genus	1699.12	0.00371	0.00379	0.98	0.3276	0.8587	
Stool NGAL	Flavonifractor	full data	Genus	219.74	-0.00454	0.00247	-1.84	0.0662	0.2122	
Stool NGAL	Flavonifractor	full data, PD/C + confounder corrected	Genus	219.74	-0.00483	0.00243	-1.98	0.0472	0.1954	
Stool NGAL	Flavonifractor	control only	Genus	186.16	0.00437	0.00337	1.29	0.1956	0.5258	
Stool NGAL	Flavonifractor	PD only	Genus	245.67	-0.01216	0.00342	-3.56	0.0004	0.0080	**
Stool NGAL	Intestinimonas	full data	Genus	97.38	-0.00509	0.00205	-2.49	0.0129	0.0694	.
Stool NGAL	Intestinimonas	full data, PD/C + confounder corrected	Genus	97.38	-0.00413	0.00196	-2.11	0.0350	0.1674	
Stool NGAL	Intestinimonas	control only	Genus	75.40	-0.00179	0.00286	-0.63	0.5300	0.8796	
Stool NGAL	Intestinimonas	PD only	Genus	113.92	-0.00820	0.00275	-2.98	0.0029	0.0314	*
Stool NGAL	Prevotella	full data	Genus	1695.01	-0.01233	0.01897	-0.65	0.5157	0.7688	
Stool NGAL	Prevotella	full data, PD/C + confounder corrected	Genus	1695.01	-0.01173	0.01901	-0.62	0.5373	0.7590	
Stool NGAL	Prevotella	control only	Genus	2608.06	-0.01719	0.00547	-3.14	0.0017	0.0294	*
Stool NGAL	Prevotella	PD only	Genus	878.85	-0.00688	0.01951	-0.35	0.7244	0.9072	
Stool NGAL	Roseburia	full data	Genus	3972.42	0.00385	0.00219	1.75	0.0794	0.2135	
Stool NGAL	Roseburia	full data, PD/C + confounder corrected	Genus	3972.42	0.00345	0.00217	1.59	0.1122	0.2990	
Stool NGAL	Roseburia	control only	Genus	5154.36	0.00737	0.00246	3.00	0.0027	0.0294	*
Stool NGAL	Roseburia	PD only	Genus	2840.63	-0.00063	0.00362	-0.17	0.8622	0.9072	
Stool NGAL	Sporobacter	full data	Genus	9.76	-0.00586	0.00188	-3.12	0.0018	0.0255	*

Stool NGAL	Sporobacter	full data, PD/C + confounder corrected	Genus	9.76	-0.00638	0.00183	-3.49	0.0005	0.0103	*
Stool NGAL	Sporobacter	control only	Genus	10.63	-0.00749	0.00280	-2.67	0.0075	0.0643	.
Stool NGAL	Sporobacter	PD only	Genus	8.86	-0.00402	0.00255	-1.58	0.1152	0.4955	
Stool NGAL	Streptococcus	full data	Genus	168.17	0.00487	0.00221	2.21	0.0272	0.1171	
Stool NGAL	Streptococcus	full data, PD/C + confounder corrected	Genus	168.17	0.00466	0.00218	2.14	0.0326	0.1674	
Stool NGAL	Streptococcus	control only	Genus	172.30	0.00141	0.00321	0.44	0.6599	0.9335	
Stool NGAL	Streptococcus	PD only	Genus	169.11	0.00808	0.00305	2.65	0.0081	0.0700	.
Stool NGAL	Veillonella	full data	Genus	63.73	0.00856	0.00329	2.60	0.0094	0.0682	.
Stool NGAL	Veillonella	full data, PD/C + confounder corrected	Genus	63.73	0.00885	0.00329	2.69	0.0072	0.0770	.
Stool NGAL	Veillonella	control only	Genus	94.64	0.00993	0.00474	2.10	0.0361	0.1747	
Stool NGAL	Veillonella	PD only	Genus	39.51	0.00424	0.00443	0.96	0.3389	0.8587	
Stool NGAL	Victivallis	full data	Genus	152.30	-0.01017	0.00376	-2.70	0.0069	0.0682	.
Stool NGAL	Victivallis	full data, PD/C + confounder corrected	Genus	152.30	-0.01747	0.01118	-1.56	0.1182	0.2990	
Stool NGAL	Victivallis	control only	Genus	178.99	-0.01193	0.00475	-2.51	0.0120	0.0736	.
Stool NGAL	Victivallis	PD only	Genus	123.51	-0.00628	0.00589	-1.07	0.2859	0.8587	
Stool zonulin	Bacteroidaceae	full data	Family	2761.00	0.00357	0.00316	1.13	0.2589	0.7572	
Stool zonulin	Bacteroidaceae	full data, PD/C + confounder corrected	Family	2761.00	0.00288	0.00315	0.91	0.3609	0.8070	
Stool zonulin	Bacteroidaceae	control only	Family	2740.99	0.01006	0.00568	1.77	0.0768	0.2646	
Stool zonulin	Bacteroidaceae	PD only	Family	2814.43	0.00039	0.00381	0.10	0.9176	0.9669	
Stool zonulin	Bifidobacteriaceae	full data	Family	3205.45	-0.00278	0.00515	-0.54	0.5893	0.8498	
Stool zonulin	Bifidobacteriaceae	full data, PD/C + confounder corrected	Family	3205.45	-0.00339	0.00466	-0.73	0.4672	0.8070	
Stool zonulin	Bifidobacteriaceae	control only	Family	1414.94	0.00967	0.00853	1.13	0.2574	0.5435	
Stool zonulin	Bifidobacteriaceae	PD only	Family	5127.66	-0.00645	0.00578	-1.12	0.2641	0.8364	
Stool zonulin	Prevotellaceae	full data	Family	1706.51	-0.00799	0.00801	-1.00	0.3188	0.7572	
Stool zonulin	Prevotellaceae	full data, PD/C + confounder corrected	Family	1706.51	-0.00606	0.03727	-0.16	0.8708	0.9026	
Stool zonulin	Prevotellaceae	control only	Family	2095.41	-0.02854	0.01248	-2.29	0.0221	0.2103	
Stool zonulin	Prevotellaceae	PD only	Family	1358.92	0.00764	0.03410	0.22	0.8228	0.9669	
Stool zonulin	Streptococcaceae	full data	Family	207.63	0.00724	0.00460	1.57	0.1156	0.7572	
Stool zonulin	Streptococcaceae	full data, PD/C + confounder corrected	Family	207.63	0.00764	0.00467	1.64	0.1016	0.8070	
Stool zonulin	Streptococcaceae	control only	Family	185.22	0.00724	0.00773	0.94	0.3489	0.6629	
Stool zonulin	Streptococcaceae	PD only	Family	217.75	0.00648	0.00568	1.14	0.2543	0.8364	
Stool zonulin	Verrucomicrobiaceae	full data	Family	698.86	0.00964	0.00681	1.42	0.1566	0.7572	
Stool zonulin	Verrucomicrobiaceae	full data, PD/C + confounder corrected	Family	698.86	0.00772	0.00689	1.12	0.2627	0.8070	
Stool zonulin	Verrucomicrobiaceae	control only	Family	702.56	0.02440	0.01150	2.12	0.0339	0.2148	
Stool zonulin	Verrucomicrobiaceae	PD only	Family	705.09	-0.00165	0.00851	-0.19	0.8462	0.9669	
Stool zonulin	Victivallaceae	full data	Family	150.36	-0.00149	0.00784	-0.19	0.8498	0.8498	
Stool zonulin	Victivallaceae	full data, PD/C + confounder corrected	Family	150.36	-0.00242	0.00794	-0.30	0.7607	0.8971	
Stool zonulin	Victivallaceae	control only	Family	168.32	-0.03884	0.01289	-3.01	0.0026	0.0490	*
Stool zonulin	Victivallaceae	PD only	Family	121.84	0.00374	0.01015	0.37	0.7128	0.9669	
Stool zonulin	Akkermansia	full data	Genus	833.65	-0.00098	0.00703	-0.14	0.8889	0.8902	
Stool zonulin	Akkermansia	full data, PD/C + confounder corrected	Genus	833.65	0.00047	0.00710	0.07	0.9469	0.9469	
Stool zonulin	Akkermansia	control only	Genus	681.42	0.01099	0.01159	0.95	0.3431	0.6761	
Stool zonulin	Akkermansia	PD only	Genus	953.40	-0.00351	0.00885	-0.40	0.6919	0.8918	
Stool zonulin	Anaerotruncus	full data	Genus	73.92	0.00455	0.00491	0.93	0.3534	0.7729	
Stool zonulin	Anaerotruncus	full data, PD/C + confounder corrected	Genus	73.92	0.00162	0.00492	0.33	0.7413	0.9042	
Stool zonulin	Anaerotruncus	control only	Genus	81.77	0.04507	0.00838	5.38	0.0000	0.0000	***
Stool zonulin	Anaerotruncus	PD only	Genus	60.74	-0.00580	0.00549	-1.06	0.2903	0.7533	
Stool zonulin	Bacteroides	full data	Genus	2746.72	0.00239	0.00316	0.76	0.4494	0.7729	
Stool zonulin	Bacteroides	full data, PD/C + confounder corrected	Genus	2746.72	0.00160	0.00309	0.52	0.6042	0.9042	
Stool zonulin	Bacteroides	control only	Genus	2717.10	0.00611	0.00570	1.07	0.2841	0.6761	
Stool zonulin	Bacteroides	PD only	Genus	2862.35	0.00071	0.00392	0.18	0.8561	0.9047	
Stool zonulin	Bifidobacterium	full data	Genus	3383.49	-0.00463	0.00522	-0.89	0.3749	0.7729	
Stool zonulin	Bifidobacterium	full data, PD/C + confounder corrected	Genus	3383.49	-0.00645	0.00465	-1.39	0.1655	0.6512	
Stool zonulin	Bifidobacterium	control only	Genus	1334.56	-0.00020	0.00820	-0.02	0.9808	0.9808	
Stool zonulin	Bifidobacterium	PD only	Genus	5469.37	-0.00827	0.00587	-1.41	0.1587	0.5632	
Stool zonulin	Coproccoccus	full data	Genus	1661.64	-0.02224	0.00480	-4.64	0.0000	0.0002	***
Stool zonulin	Coproccoccus	full data, PD/C + confounder corrected	Genus	1661.64	-0.02257	0.00484	-4.66	0.0000	0.0001	***
Stool zonulin	Coproccoccus	control only	Genus	1665.85	-0.02265	0.00795	-2.85	0.0044	0.0376	*
Stool zonulin	Coproccoccus	PD only	Genus	1657.24	-0.02294	0.00605	-3.79	0.0001	0.0064	**
Stool zonulin	Flavonifractor	full data	Genus	212.39	0.00784	0.00494	1.59	0.1126	0.6051	
Stool zonulin	Flavonifractor	full data, PD/C + confounder corrected	Genus	212.39	0.00759	0.00491	1.54	0.1224	0.6512	
Stool zonulin	Flavonifractor	control only	Genus	181.49	0.03436	0.00821	4.19	0.0000	0.0004	***
Stool zonulin	Flavonifractor	PD only	Genus	239.89	-0.00108	0.00602	-0.18	0.8581	0.9047	
Stool zonulin	Intestinimonas	full data	Genus	95.18	-0.00205	0.00424	-0.48	0.6284	0.8616	
Stool zonulin	Intestinimonas	full data, PD/C + confounder corrected	Genus	95.18	-0.00069	0.00408	-0.17	0.8655	0.9136	

Stool zonulin	Intestinimonas	control only	Genus	73.57	0.01238	0.00739	1.68	0.0939	0.5047	
Stool zonulin	Intestinimonas	PD only	Genus	111.58	-0.00728	0.00489	-1.49	0.1363	0.5632	
Stool zonulin	Prevotella	full data	Genus	1782.65	-0.01162	0.03925	-0.30	0.7672	0.8800	
Stool zonulin	Prevotella	full data, PD/C + confounder corrected	Genus	1782.65	-0.01164	0.03958	-0.29	0.7686	0.9042	
Stool zonulin	Prevotella	control only	Genus	2646.55	-0.07115	0.01371	-5.19	0.0000	0.0000	***
Stool zonulin	Prevotella	PD only	Genus	1010.27	0.00776	0.03411	0.23	0.8200	0.9047	
Stool zonulin	Roseburia	full data	Genus	3994.13	0.00541	0.00445	1.22	0.2236	0.7069	
Stool zonulin	Roseburia	full data, PD/C + confounder corrected	Genus	3994.13	0.00767	0.00439	1.75	0.0809	0.6512	
Stool zonulin	Roseburia	control only	Genus	5167.52	0.00175	0.00645	0.27	0.7856	0.9061	
Stool zonulin	Roseburia	PD only	Genus	2885.18	0.01120	0.00588	1.90	0.0570	0.3843	
Stool zonulin	Sporobacter	full data	Genus	9.45	-0.00521	0.00403	-1.29	0.1956	0.7069	
Stool zonulin	Sporobacter	full data, PD/C + confounder corrected	Genus	9.45	-0.00418	0.00408	-1.02	0.3064	0.7814	
Stool zonulin	Sporobacter	control only	Genus	10.56	-0.01116	0.00804	-1.39	0.1649	0.6761	
Stool zonulin	Sporobacter	PD only	Genus	8.46	-0.00421	0.00451	-0.93	0.3504	0.7533	
Stool zonulin	Streptococcus	full data	Genus	214.27	0.00588	0.00495	1.19	0.2346	0.7069	
Stool zonulin	Streptococcus	full data, PD/C + confounder corrected	Genus	214.27	0.00489	0.00499	0.98	0.3271	0.7814	
Stool zonulin	Streptococcus	control only	Genus	171.12	-0.00053	0.00819	-0.07	0.9482	0.9707	
Stool zonulin	Streptococcus	PD only	Genus	251.15	0.00592	0.00632	0.94	0.3489	0.7533	
Stool zonulin	Veillonella	full data	Genus	91.54	0.00884	0.00716	1.23	0.2168	0.7069	
Stool zonulin	Veillonella	full data, PD/C + confounder corrected	Genus	91.54	0.01034	0.00716	1.44	0.1486	0.6512	
Stool zonulin	Veillonella	control only	Genus	100.50	-0.01019	0.03764	-0.27	0.7866	0.9061	
Stool zonulin	Veillonella	PD only	Genus	84.87	0.01143	0.00833	1.37	0.1703	0.5632	
Stool zonulin	Victivallis	full data	Genus	160.22	-0.00160	0.00787	-0.20	0.8391	0.8800	
Stool zonulin	Victivallis	full data, PD/C + confounder corrected	Genus	160.22	-0.00286	0.00798	-0.36	0.7196	0.9042	
Stool zonulin	Victivallis	control only	Genus	176.22	-0.04673	0.01277	-3.66	0.0003	0.0027	**
Stool zonulin	Victivallis	PD only	Genus	142.57	0.00402	0.01001	0.40	0.6877	0.8918	
Stool calprotectin	Bacteroidaceae	full data	Family	2765.44	-0.00003	0.00009	-0.40	0.6888	0.8601	
Stool calprotectin	Bacteroidaceae	full data, PD/C + confounder corrected	Family	2765.44	-0.00001	0.00009	-0.12	0.9021	0.9523	
Stool calprotectin	Bacteroidaceae	control only	Family	2764.47	-0.00002	0.00018	-0.10	0.9237	0.9237	
Stool calprotectin	Bacteroidaceae	PD only	Family	2814.43	-0.00008	0.00010	-0.76	0.4469	0.7719	
Stool calprotectin	Bifidobacteriaceae	full data	Family	3207.98	0.00013	0.00014	0.91	0.3651	0.6843	
Stool calprotectin	Bifidobacteriaceae	full data, PD/C + confounder corrected	Family	3207.98	-0.00005	0.00013	-0.38	0.7047	0.9523	
Stool calprotectin	Bifidobacteriaceae	control only	Family	1388.87	-0.00037	0.00026	-1.43	0.1535	0.6106	
Stool calprotectin	Bifidobacteriaceae	PD only	Family	5127.66	0.00002	0.00015	0.14	0.8899	0.8899	
Stool calprotectin	Prevotellaceae	full data	Family	1713.63	0.00018	0.00101	0.18	0.8601	0.8601	
Stool calprotectin	Prevotellaceae	full data, PD/C + confounder corrected	Family	1713.63	0.00031	0.00105	0.30	0.7660	0.9523	
Stool calprotectin	Prevotellaceae	control only	Family	2138.13	0.00019	0.00039	0.47	0.6356	0.8919	
Stool calprotectin	Prevotellaceae	PD only	Family	1358.92	0.00026	0.00090	0.29	0.7700	0.8899	
Stool calprotectin	Streptococcaceae	full data	Family	194.59	0.00048	0.00011	4.26	0.0000	0.0004	***
Stool calprotectin	Streptococcaceae	full data, PD/C + confounder corrected	Family	194.59	0.00053	0.00012	4.59	0.0000	0.0001	***
Stool calprotectin	Streptococcaceae	control only	Family	157.57	0.00013	0.00023	0.58	0.5638	0.8919	
Stool calprotectin	Streptococcaceae	PD only	Family	217.75	0.00061	0.00013	4.88	0.0000	0.0000	***
Stool calprotectin	Verrucomicrobiaceae	full data	Family	701.07	0.00037	0.00018	2.00	0.0450	0.2137	
Stool calprotectin	Verrucomicrobiaceae	full data, PD/C + confounder corrected	Family	701.07	0.00040	0.00019	2.09	0.0364	0.2302	
Stool calprotectin	Verrucomicrobiaceae	control only	Family	701.00	0.00070	0.00035	2.03	0.0423	0.4015	
Stool calprotectin	Verrucomicrobiaceae	PD only	Family	705.09	0.00005	0.00023	0.21	0.8331	0.8899	
Stool calprotectin	Victivallaceae	full data	Family	149.22	0.00004	0.00021	0.18	0.8540	0.8601	
Stool calprotectin	Victivallaceae	full data, PD/C + confounder corrected	Family	149.22	-0.00016	0.00067	-0.24	0.8094	0.9523	
Stool calprotectin	Victivallaceae	control only	Family	166.11	-0.00028	0.00040	-0.69	0.4876	0.8422	
Stool calprotectin	Victivallaceae	PD only	Family	121.84	0.00030	0.00027	1.14	0.2556	0.5509	
Stool calprotectin	Akkermansia	full data	Genus	835.84	0.00029	0.00019	1.54	0.1229	0.4805	
Stool calprotectin	Akkermansia	full data, PD/C + confounder corrected	Genus	835.84	0.00035	0.00020	1.77	0.0767	0.3297	
Stool calprotectin	Akkermansia	control only	Genus	687.58	0.00045	0.00035	1.28	0.2009	0.8121	
Stool calprotectin	Akkermansia	PD only	Genus	953.40	0.00024	0.00023	1.05	0.2943	0.7659	
Stool calprotectin	Anaerotruncus	full data	Genus	74.15	-0.00016	0.00013	-1.20	0.2292	0.6710	
Stool calprotectin	Anaerotruncus	full data, PD/C + confounder corrected	Genus	74.15	-0.00021	0.00014	-1.51	0.1309	0.4330	
Stool calprotectin	Anaerotruncus	control only	Genus	82.97	-0.00037	0.00027	-1.35	0.1774	0.8121	
Stool calprotectin	Anaerotruncus	PD only	Genus	60.74	-0.00004	0.00015	-0.30	0.7652	0.9522	
Stool calprotectin	Bacteroides	full data	Genus	2753.57	0.00005	0.00009	0.53	0.5952	0.8381	
Stool calprotectin	Bacteroides	full data, PD/C + confounder corrected	Genus	2753.57	0.00005	0.00008	0.63	0.5295	0.9129	
Stool calprotectin	Bacteroides	control only	Genus	2728.22	0.00002	0.00018	0.10	0.9209	0.9817	
Stool calprotectin	Bacteroides	PD only	Genus	2862.35	0.00004	0.00010	0.36	0.7170	0.9522	
Stool calprotectin	Bifidobacterium	full data	Genus	3367.85	0.00022	0.00014	1.59	0.1122	0.4805	
Stool calprotectin	Bifidobacterium	full data, PD/C + confounder corrected	Genus	3367.85	0.00002	0.00013	0.15	0.8830	0.9753	
Stool calprotectin	Bifidobacterium	control only	Genus	1310.24	-0.00030	0.00025	-1.20	0.2294	0.8121	
Stool calprotectin	Bifidobacterium	PD only	Genus	5469.37	0.00010	0.00016	0.64	0.5238	0.8233	

Stool calprotectin	Coprococcus	full data	Genus	1653.22	0.00012	0.00014	0.86	0.3921	0.7331	
Stool calprotectin	Coprococcus	full data, PD/C + confounder corrected	Genus	1653.22	0.00007	0.00014	0.48	0.6310	0.9357	
Stool calprotectin	Coprococcus	control only	Genus	1647.24	0.00000	0.00026	-0.01	0.9954	0.9954	
Stool calprotectin	Coprococcus	PD only	Genus	1657.24	0.00016	0.00017	0.97	0.3314	0.7659	
Stool calprotectin	Flavonifractor	full data	Genus	213.13	0.00009	0.00013	0.68	0.4993	0.7668	
Stool calprotectin	Flavonifractor	full data, PD/C + confounder corrected	Genus	213.13	0.00003	0.00014	0.21	0.8354	0.9753	
Stool calprotectin	Flavonifractor	control only	Genus	182.80	-0.00015	0.00027	-0.56	0.5728	0.8827	
Stool calprotectin	Flavonifractor	PD only	Genus	239.89	0.00013	0.00016	0.80	0.4255	0.7761	
Stool calprotectin	Intestinimonas	full data	Genus	95.46	-0.00028	0.00011	-2.50	0.0125	0.1078	
Stool calprotectin	Intestinimonas	full data, PD/C + confounder corrected	Genus	95.46	-0.00031	0.00011	-2.81	0.0049	0.0700	
Stool calprotectin	Intestinimonas	control only	Genus	74.44	-0.00033	0.00023	-1.47	0.1421	0.8121	
Stool calprotectin	Intestinimonas	PD only	Genus	111.58	-0.00033	0.00013	-2.60	0.0093	0.1328	
Stool calprotectin	Prevotella	full data	Genus	1785.99	0.00020	0.00105	0.19	0.8492	0.9651	
Stool calprotectin	Prevotella	full data, PD/C + confounder corrected	Genus	1785.99	0.00033	0.00109	0.30	0.7612	0.9753	
Stool calprotectin	Prevotella	control only	Genus	2679.55	0.00056	0.00146	0.38	0.7014	0.9274	
Stool calprotectin	Prevotella	PD only	Genus	1010.27	0.00028	0.00090	0.31	0.7600	0.9522	
Stool calprotectin	Roseburia	full data	Genus	3987.84	0.00009	0.00012	0.71	0.4790	0.7668	
Stool calprotectin	Roseburia	full data, PD/C + confounder corrected	Genus	3987.84	0.00014	0.00012	1.12	0.2619	0.6256	
Stool calprotectin	Roseburia	control only	Genus	5187.27	0.00014	0.00020	0.70	0.4834	0.8726	
Stool calprotectin	Roseburia	PD only	Genus	2885.18	0.00016	0.00016	0.99	0.3199	0.7659	
Stool calprotectin	Sporobacter	full data	Genus	9.48	-0.00023	0.00011	-2.17	0.0299	0.1834	
Stool calprotectin	Sporobacter	full data, PD/C + confounder corrected	Genus	9.48	-0.00021	0.00011	-1.95	0.0515	0.2597	
Stool calprotectin	Sporobacter	control only	Genus	10.67	-0.00023	0.00023	-0.99	0.3223	0.8121	
Stool calprotectin	Sporobacter	PD only	Genus	8.46	-0.00023	0.00012	-1.94	0.0526	0.3980	
Stool calprotectin	Streptococcus	full data	Genus	203.21	0.00057	0.00012	4.78	0.0000	0.0001	***
Stool calprotectin	Streptococcus	full data, PD/C + confounder corrected	Genus	203.21	0.00059	0.00012	4.78	0.0000	0.0001	***
Stool calprotectin	Streptococcus	control only	Genus	147.14	0.00001	0.00024	0.05	0.9588	0.9817	
Stool calprotectin	Streptococcus	PD only	Genus	251.15	0.00070	0.00014	5.06	0.0000	0.0000	***
Stool calprotectin	Veillonella	full data	Genus	75.75	0.00068	0.00017	3.93	0.0001	0.0019	**
Stool calprotectin	Veillonella	full data, PD/C + confounder corrected	Genus	75.75	0.00079	0.00018	4.35	0.0000	0.0003	***
Stool calprotectin	Veillonella	control only	Genus	65.54	0.00103	0.00035	2.96	0.0031	0.0817	.
Stool calprotectin	Veillonella	PD only	Genus	84.87	0.00059	0.00020	2.91	0.0036	0.0779	.
Stool calprotectin	Victivallis	full data	Genus	159.49	0.00010	0.00021	0.46	0.6489	0.8381	
Stool calprotectin	Victivallis	full data, PD/C + confounder corrected	Genus	159.49	-0.00014	0.00070	-0.20	0.8421	0.9753	
Stool calprotectin	Victivallis	control only	Genus	173.24	-0.00042	0.00038	-1.09	0.2746	0.8121	
Stool calprotectin	Victivallis	PD only	Genus	142.57	0.00033	0.00026	1.26	0.2086	0.7659	

Tables include the results for families and genera that had a multiple comparison corrected $p < 0.05$ in at least one comparison for A) Acetic, propionic, and butyric acid or B) Stool NGAL, zonulin, and calprotectin.

R analyses on Parkinson's disease, gut microbiota, inflammatory markers and short-chain fatty acids

Velma T. E. Aho

29 May 2020

Contents

1	Setup	2
1.1	Data import	2
1.2	Outlier check	4
1.3	Subjects and demographics	6
2	Comparisons without microbiota data	13
2.1	Intercorrelations of inflammatory markers and SCFAs	13
2.2	PCA with inflammatory markers	16
2.2.1	Stool	16
2.2.2	Plasma	19
2.3	Correlations between clinical variables	21
2.3.1	SCFAs, markers vs other variables	22
3	Comparisons with microbiota data	36
3.1	Alpha diversity	38
3.1.1	Full data (PD + C together)	38
3.1.2	Control only / PD only	39
3.1.3	Corrected for PD vs C	41
3.1.4	Analyses with additional confounders	44
3.2	Beta diversity	50
3.2.1	Full data (PD + C together)	50
3.2.2	Corrected for PD vs control	51
3.2.3	Control only / PD only	52
3.2.4	Analyses with additional confounders	53
3.2.5	Interaction effects	56
3.3	Enterotypes	58
3.4	Differential abundance with DESeq2	61
3.4.1	SCFAs	62
3.4.2	Inflammatory markers	68
4	Session info	75

1 Setup

```
# Load required packages
options(kableExtra.latex.load_packages=FALSE)
library("kableExtra")
library("ggplot2")
library("gridExtra")
library("reshape2")
library("Hmisc")
library("factoextra")
library("gtools")
library("ggcorrplot")
library("GGally")
library("phyloseq")
library("vegan")
library("DESeq2")

# Dimensions for figures:
maxwd <- 190
maxhi <- 225
```

1.1 Data import

Import inflammatory marker data, mark as NA the values that are over the measurement range for calprotectin and NGAL, and trim subjects that only have values for LBP:

```
inf_df <- read.csv("Inputs/cytokine_data.csv", row.names = 1)

# Save the variable names for further use:
inf_vars <- colnames(inf_df)
inf_vars_s <- grep("^S", inf_vars, value = TRUE) # Stool variables
inf_vars_p <- grep("^p", inf_vars, value = TRUE) # Plasma variables

# Change the "TOO HIGH" values in stool NGAL to NA
inf_df$$SNGAL <- as.character(inf_df$$SNGAL)
inf_df$$SNGAL[inf_df$$SNGAL == "TOO HIGH"] <- NA
inf_df$$SNGAL <- as.numeric(inf_df$$SNGAL)

# Change the "TOO HIGH" values in stool calprotectin to NA
inf_df$$Scalprotectin <- as.character(inf_df$$Scalprotectin)
inf_df$$Scalprotectin[inf_df$$Scalprotectin == "TOO HIGH"] <- NA
inf_df$$Scalprotectin <- as.numeric(inf_df$$Scalprotectin)

# Samples with more than one missing value?
print(inf_df[names(which(rowSums(is.na(inf_df)) > 1)), ])
```

```
##           pLBP SNGAL Scalprotectin Szonulin SIFNg SIL10 SIL12p70 SIL13 SIL1b
## C0052 4.653704    NA              NA      NA    NA    NA      NA    NA    NA
## C0124 4.283837    NA              NA      NA    NA    NA      NA    NA    NA
##           SIL2 SIL4 SIL6 SCXCL8 STNF pIFNg pIL10 pIL12p70 pIL13 pIL1b pIL2 pIL4
## C0052    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
## C0124    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
##           pIL6 pCXCL8 pTNF
## C0052    NA      NA    NA
```



```
## C0124    NA      NA    NA
```

```
# Two samples have NA for everything except pLBP. Drop these two:
```

```
inf_df <- inf_df[~(which(rowSums(is.na(inf_df)) > 1)), ]
```

Import clinical metadata and SCFA measurements, trim to subjects fit to use based on the follow-up analyses (Aho et al. 2019):

```
clin_df <- read.csv("Inputs/pdcy_meta_v4.csv", row.names = 1)
```

```
# Subset to follow-up timepoint samples deemed "OK to use"
```

```
clin_df <- subset(clin_df, Timepoint == "followup" & OK_to_use == "yes")
```

```
# Remove timepoint identifier from row names
```

```
rownames(clin_df) <- sub("N", "", rownames(clin_df))
```

```
# Make sure factor variables are in correct form
```

```
for(i in colnames(clin_df)){  
  if(identical(names(table(clin_df[, i])), c("0", "1"))) | is.factor(clin_df[, i])){  
    clin_df[, i] <- factor(clin_df[, i])  
  }  
}
```

```
# Two additional variables need to be fixed manually:
```

```
clin_df$tobacco_when_last <- factor(clin_df$tobacco_when_last)
```

```
clin_df$meds_rasagiline_mg <- as.numeric(as.character(clin_df$meds_rasagiline_mg))
```

```
# Remove some redundant variables
```

```
clin_df$OK_to_use <- NULL
```

```
clin_df$Timepoint <- NULL
```

```
# Remove variables with a lot of missing values
```

```
clin_df <- clin_df[, ~which(colSums(is.na(clin_df)) > 100)]
```

```
# Remove variables with a lot of zeros
```

```
clin_df <- clin_df[, ~which(colSums(clin_df == 0, na.rm = TRUE) > 120)]
```

Import enterotype variable and add to clinical data:

```
pd_et <- read.csv("Inputs/enterotyping_results.txt", row.names = 1)
```

```
# Trim to follow-up only
```

```
pd_et <- pd_et[grep("N", rownames(pd_et)), ]
```

```
rownames(pd_et) <- sub("N", "", rownames(pd_et))
```

```
# Add to main metadata df
```

```
clin_df$Enterotype <- pd_et[rownames(clin_df), "ET"]
```

Trim subjects with missing values for any of the big three SCFAs:

```
# Make a list of the SCFA variables
```

```
scfa_vars <- grep("_acid", colnames(clin_df), value = TRUE)
```

```
# Reorder for convenience
```

```
scfa_vars <- scfa_vars[c(1, 2, 4, 3, 5:6)]
```

```
# Exclude subjects with missing values for any of the big 3 SCFAs
```

```
clin_df <- subset(clin_df, !rowSums(is.na(clin_df[, scfa_vars[1:3]])) > 0)
```

Add a "total SCFAs" sum variable:

```
clin_df$total_SCFAs <- rowSums(clin_df[, scfa_vars], na.rm = TRUE)
scfa_vars <- c("total_SCFAs", scfa_vars)
```

Combine the immune marker data and the rest of the metadata:

```
# Match the immune marker data to the trimmed clinical data
inf_df <- inf_df[rownames(clin_df), ]

# Combine the two dfs
pdcy_meta <- cbind(clin_df, inf_df)
```

1.2 Outlier check

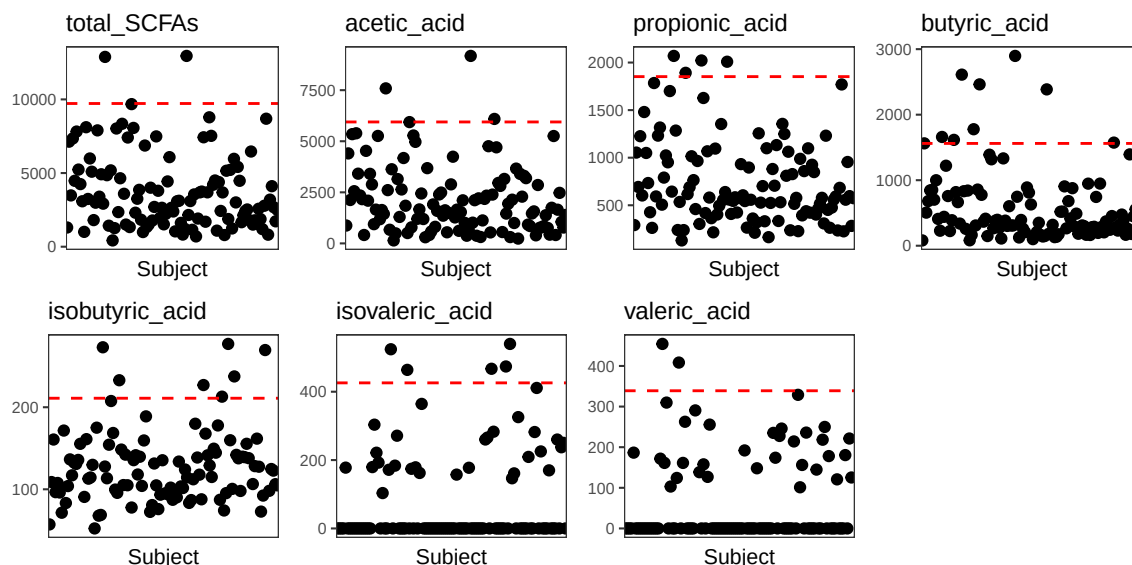
Plot values of the variables of interest in all samples to see how they are distributed.

1.2.0.1 SCFAs

Plot all SCFA measurement values (dashed line is the 3rd quartile + 1.5*IQR):

```
# Plotting function
ol_plot <- function(df, var){
  ggplot(df, aes_string(x = "Subject", y = var)) +
    geom_point() +
    theme_bw(base_size = 8) +
    ylab(NULL) +
    ggtitle(var) +
    geom_hline(yintercept = (quantile(df[,var], 0.75, na.rm = TRUE) +
                             1.5*IQR(df[,var], na.rm = TRUE)),
              linetype = 2, color = "red") +
    theme(panel.grid = element_blank(),
          axis.text.x = element_blank(),
          axis.ticks.x = element_blank(),
          plot.title = element_text(size = 9))
}

scfa_ol <- lapply(scfa_vars, function(x) ol_plot(pdcy_meta, x))
do.call("grid.arrange", c(scfa_ol, ncol = 4))
```



The SCFA data seemed quite evenly distributed. Since most subjects' value for valeric and isovaleric acid was zero, convert these two into binary variables (0 vs > 0):

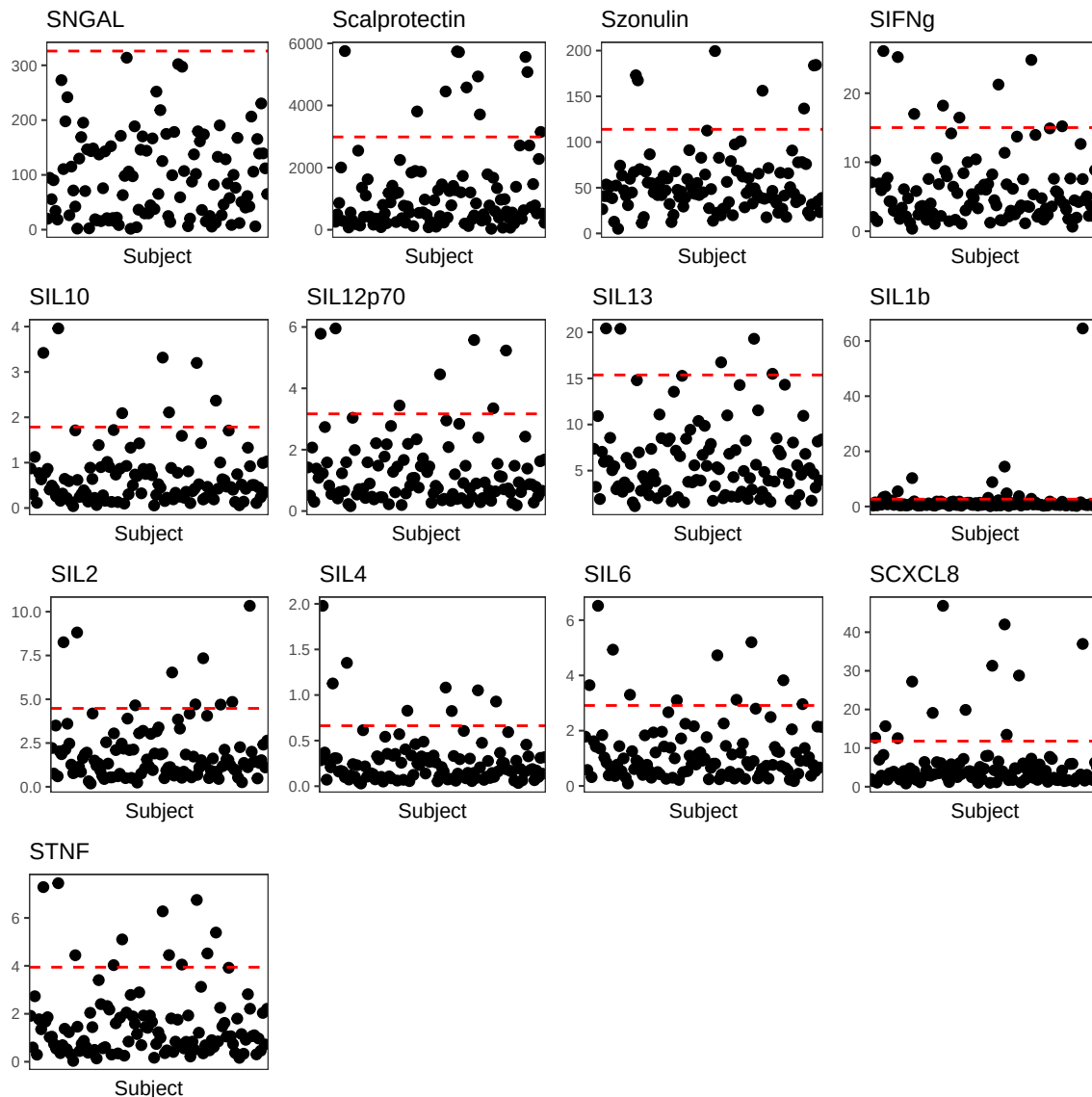
```
# Make new binary variables for the two SCFAs
pdcy_meta$isovaleric_acid_binary <- as.numeric(pdcy_meta$isovaleric_acid > 0)
pdcy_meta$valeric_acid_binary <- as.numeric(pdcy_meta$isovaleric_acid > 0)

# Replace the numeric variables in the list of SCFA vars with the binary ones
scfa_vars[grepl("valeric", scfa_vars)] <- paste(scfa_vars[grepl("valeric", scfa_vars)],
                                                "binary", sep = "_")
```

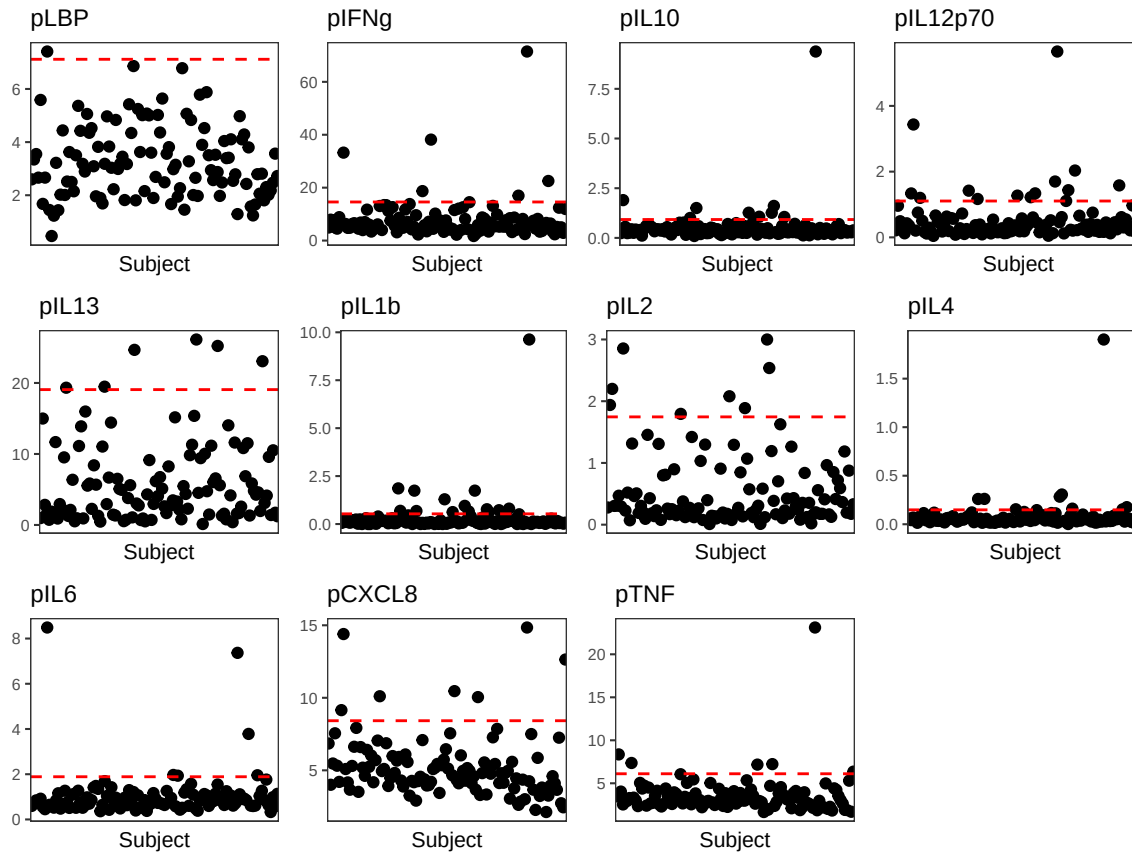
1.2.0.2 Inflammatory markers

Plot all inflammatory marker values:

```
inf_ol <- lapply(c(inf_vars_s, inf_vars_p), function(x) ol_plot(pdcy_meta, x))
do.call("grid.arrange", c(inf_ol[1:13], ncol = 4))
```



```
do.call("grid.arrange", c(inf_ol[14:24], ncol = 4))
```



There was a single, extreme outlier point in pIL1b, pIL10, pIL1b, pIL4 and pTNF. The subjects of these points were:

```
sapply(c("pIL1b", "pIL10", "pIL1b", "pIL4", "pTNF"), function(x)
  as.vector(pdcy_meta[which(pdcy_meta[, x] == max(pdcy_meta[, x])), "Subject"]))
```

```
##   pIL1b   pIL10   pIL1b   pIL4   pTNF
## "P0099" "P0071" "P0071" "P0071" "P0071"
```

For plasma measurements, the outlier was the same subject. To make sure this subject would not cause problems, exclude them from the data. Replace the one outlier pIL1b value with NA.

```
pdcy_meta[pdcy_meta$Subject == "P0099", "pIL1b"] <- NA
pdcy_meta <- subset(pdcy_meta, Subject != "P0071")
```

1.3 Subjects and demographics

After the exclusions, the number of samples remaining for the analyses was as follows:

```
kable(table(pdcy_meta$Group), col.names = c("Group", "n"), booktabs = TRUE)
```

Group	n
C	56
P	55

```
# Export the final metadata for possible future use
write.csv(pdcy_meta, "Outputs/pdcy_meta_from_R.csv")
```

1.3.0.1 Table 1

Make tables of basic demographics and measures for the samples:

Categorical confounders

```
# List all binary variables
clin_table_cat_vars <- names(which(lengths(sapply(pdcy_meta, levels)) == 2))
# Exclude PD vs C
clin_table_cat_vars <- grep("Group", clin_table_cat_vars, invert = TRUE, value = TRUE)

# Summarize
clin_table_cat <- sapply(clin_table_cat_vars,
  function(x){
    out <- c(paste(levels(pdcy_meta[, x])[2]),
      prop.table(
        table(pdcy_meta[,c("Group", x)]), 1)[1,2]*100,
      prop.table(
        table(pdcy_meta[,c("Group", x)]), 1)[2,2]*100,
      fisher.test(
        table(pdcy_meta[,c("Group", x)])$p.value)
      names(out) <- c("var", "C", "P", "pVal")
      return(out)})

# Reorganize
clin_table_cat <- as.data.frame(t(clin_table_cat))
clin_table_cat$var <- capitalize(
  gsub("_", " ", paste0(rownames(clin_table_cat), " (% ",
    sub(1, "yes", clin_table_cat$var), ")"))))
clin_table_cat[, 2:4] <- sapply(clin_table_cat[, 2:4], function(x)
  as.numeric(as.character(x)))

# Trim to significant variables + sex
clin_table_cat_sig <- rbind(clin_table_cat["sex",],
  subset(clin_table_cat, pVal < 0.05))

# Show
kable_styling(kable(clin_table_cat_sig, digits = 4,
  booktabs = TRUE, row.names = FALSE,
  linesep = ""), font_size = 8)
```

var	C	P	pVal
Sex (% M)	48.2143	52.7273	0.7056
History TIA ischemic stroke (% yes)	37.5000	5.5556	0.0000
Meds ca antagonist (% yes)	19.6429	5.4545	0.0424
Meds COMT inhibitor (% yes)	0.0000	27.2727	0.0000
Meds dopa (% yes)	0.0000	76.3636	0.0000
Meds dopamine agonist (% yes)	0.0000	80.0000	0.0000
Meds MAO inhibitor (% yes)	0.0000	70.9091	0.0000
Meds statin (% yes)	48.2143	18.1818	0.0012
Rome III IBS criteria fulfilled (% yes)	7.1429	36.3636	0.0002
Rome III IBS possible (% yes)	7.1429	38.1818	0.0001

```
# Export full results
write.csv(clin_table_cat, "Outputs/table1cat.csv", row.names = FALSE)
```

Numeric confounders

```
# Function for summarizing
meanSdP <- function(df, x){
  out <- c(
    paste0(
      round(mean(subset(df, Group == "C")[, x], na.rm = TRUE), digits = 2), " ± ",
      round(sd(subset(df, Group == "C")[, x], na.rm = TRUE), digits = 2)),
    paste0(
      round(mean(subset(df, Group == "P")[, x], na.rm = TRUE), digits = 2), " ± ",
      round(sd(subset(df, Group == "P")[, x], na.rm = TRUE), digits = 2)),
      round(wilcox.test(as.formula(paste(x, "~ Group")), df)$p.value, digits = 4))
  names(out) <- c("C", "P", "pVal")
  return(out)
}

# Grab all numeric variables
clin_table_num_vars <- colnames(pdcy_meta)[sapply(
  pdcy_meta, class) %in% c("numeric", "integer")]
# Exclude variables of interest
clin_table_num_vars <- clin_table_num_vars[1:(grep("acid", clin_table_num_vars)[1]-1)]
# Drop some PD-specific variables
clin_table_num_vars <- clin_table_num_vars[!(clin_table_num_vars %in% c(
  grep("onset", clin_table_num_vars, value = TRUE),
  grep("jankovic", clin_table_num_vars, value = TRUE),
  grep("poletti", clin_table_num_vars, value = TRUE),
  grep("UPDRS", clin_table_num_vars, value = TRUE)))]

# Summarize & rearrange
clin_table_num <- sapply(clin_table_num_vars, function(x) meanSdP(pdcy_meta, x))
clin_table_num <- as.data.frame(t(clin_table_num))
clin_table_num$pVal <- as.numeric(as.character(clin_table_num$pVal))
clin_table_num <- data.frame(var = capitalize(
  gsub("_", " ", rownames(clin_table_num))), clin_table_num)

# Show
kable_styling(kable(clin_table_num, booktabs = TRUE, row.names = FALSE,
  linesep = ""), font_size = 8)
```

var	C	P	pVal
Age at stool collection	66.38 ± 6.73	67.63 ± 5.21	0.2930
BMI	26.7 ± 3.56	27.63 ± 4.74	0.2612
GDS15	0.84 ± 1.39	4.36 ± 3.72	0.0000
LED	0 ± 0	608.61 ± 301.46	0.0000
Meds entacapone mg	0 ± 0	238.18 ± 413.88	0.0000
Meds levodopa entacapone mg	0 ± 0	130.45 ± 243.94	0.0000
Meds levodopa IR mg	0 ± 0	152.73 ± 209.81	0.0000
Meds pramipexole mg	0 ± 0	0.6 ± 0.9	0.0000
Meds rasagiline mg	0 ± 0	0.36 ± 0.49	0.0000
Meds ropinirole mg	0 ± 0	3.05 ± 5.76	0.0000
Meds selegiline mg	0 ± 0	3.19 ± 4.64	0.0000
MMSE total	28.82 ± 1.49	28.15 ± 1.7	0.0160
NMSQuest total	2.45 ± 2.1	10.91 ± 5.14	0.0000
NMSS total	7.09 ± 6.35	48.36 ± 36.55	0.0000
RBDSQ	1.73 ± 1.29	5.07 ± 3.07	0.0000
RLS	0.16 ± 0.63	1.64 ± 1.77	0.0000
Rome III constip defec sumscore 9.15	2.46 ± 3.16	7.45 ± 5.22	0.0000
SCS PD total	0.18 ± 0.47	4.38 ± 4.24	0.0000
SDQ total	0.5 ± 0.89	5.73 ± 4.82	0.0000
Sniffinsticks	13.52 ± 1.55	7.35 ± 2.95	0.0000
Wexner total	2.2 ± 2.14	6.38 ± 4.69	0.0000

```
# Export full results
write.csv(clin_table_num, "Outputs/table1num.csv", row.names = FALSE)
```

1.3.0.2 Table S1

Descriptive values and *p*-values for PD/control and the variables of interest (SCFAs + inflammatory markers):

```
# List variables of interest (with isovaleric and valeric acid as numeric)
basic_pdc_vars <- c(scfa_vars[1:5], "isovaleric_acid", "valeric_acid",
  inf_vars_s, inf_vars_p)

# Summarize
table_basic_pdc <- sapply(basic_pdc_vars, function(x) meanSdP(pdcy_meta, x))

# Function for fixing variable names
varLabelFix <- function(x){
  capitalize(
    gsub("_", " ", sub("^p([A-Z]{1})", "Plasma \\1", sub("^S", "Stool ", x)))
  )
}

# Reorganize and make nicer row labels
table_basic_pdc <- as.data.frame(t(table_basic_pdc))
table_basic_pdc <- data.frame(var = varLabelFix(rownames(table_basic_pdc)),
  table_basic_pdc)

# Show
kable_styling(kable(table_basic_pdc, booktabs = TRUE, row.names = FALSE,
  linesep = ""), font_size = 8)
```

var	C	P	pVal
Total SCFAs	4209.14 ± 2688.93	3288.72 ± 2388.22	0.0506
Acetic acid	2390.65 ± 1699.99	1938.61 ± 1669.64	0.0922
Propionic acid	840.13 ± 509.61	644.38 ± 351.39	0.0422
Butyric acid	736.26 ± 633.78	424.61 ± 405.92	0.0013
Isobutyric acid	123.99 ± 43.24	131.02 ± 46.37	0.6149
Isovaleric acid	65.59 ± 125.07	92.63 ± 149.41	0.4013
Valeric acid	60.24 ± 113.12	65.82 ± 99.15	0.5337
Stool NGAL	93.88 ± 78.46	100.92 ± 74.89	0.5313
Stool calprotectin	872.01 ± 1000.65	1499.67 ± 1616.72	0.0233
Stool zonulin	52.86 ± 30.38	58.53 ± 42.73	0.8066
Stool IFNg	6.35 ± 5.5	5.68 ± 4.8	0.5416
Stool IL10	0.73 ± 0.73	0.67 ± 0.69	0.4846
Stool IL12p70	1.36 ± 1.17	1.13 ± 1.06	0.1778
Stool IL13	6.26 ± 4.32	5.73 ± 3.99	0.4773
Stool IL1b	1.24 ± 1.53	1.38 ± 2.28	0.505
Stool IL2	1.98 ± 1.69	1.9 ± 1.88	0.4809
Stool IL4	0.31 ± 0.34	0.25 ± 0.24	0.2818
Stool IL6	1.29 ± 1.18	1.15 ± 1.05	0.3779
Stool CXCL8	5.82 ± 7.53	5.96 ± 8.6	0.5299
Stool TNF	1.63 ± 1.56	1.54 ± 1.51	0.6434
Plasma LBP	3.38 ± 1.45	3.21 ± 1.26	0.5455
Plasma IFNg	8.04 ± 6.33	6.77 ± 4.04	0.2513
Plasma IL10	0.45 ± 0.31	0.45 ± 0.3	0.7975
Plasma IL12p70	0.46 ± 0.53	0.58 ± 0.82	0.2637
Plasma IL13	5.35 ± 5.64	6.48 ± 5.99	0.1188
Plasma IL1b	0.23 ± 0.38	0.21 ± 0.31	0.906
Plasma IL2	0.56 ± 0.64	0.56 ± 0.61	0.6518
Plasma IL4	0.06 ± 0.05	0.07 ± 0.06	0.3747
Plasma IL6	0.96 ± 1.07	0.99 ± 0.55	0.1816
Plasma CXCL8	5.46 ± 1.87	4.81 ± 2.03	0.008
Plasma TNF	3.53 ± 1.23	3.25 ± 1.31	0.1119

1.3.0.3 Table S1: Split by sex

Check for differences between PD and control when data is split to male or female.

```
# Summarize F-only and M-only subsets
table_basic_pdc_F <- sapply(basic_pdc_vars, function(x)
  meanSDP(subset(pdcy_meta, sex == "F"), x))
table_basic_pdc_M <- sapply(basic_pdc_vars, function(x)
  meanSDP(subset(pdcy_meta, sex == "M"), x))

# Reorganize and make nicer row labels
table_basic_pdc_F <- as.data.frame(t(table_basic_pdc_F))
colnames(table_basic_pdc_F) <- paste(colnames(table_basic_pdc_F),
  "female", sep = "_")
table_basic_pdc_F$pVal_female <- as.numeric(
  as.character(table_basic_pdc_F$pVal_female))

table_basic_pdc_M <- as.data.frame(t(table_basic_pdc_M))
colnames(table_basic_pdc_M) <- paste(colnames(table_basic_pdc_M),
  "male", sep = "_")
table_basic_pdc_M$pVal_male <- as.numeric(
  as.character(table_basic_pdc_M$pVal_male))

# Combine dfs
table_basic_pdc_sex <- cbind(table_basic_pdc_F, table_basic_pdc_M)
```



```
# Fix rownames
table_basic_pdc_sex <- data.frame(var = varLabelFix(rownames(table_basic_pdc_sex)),
                                  table_basic_pdc_sex)

# Significant results?
kable_styling(kable(subset(table_basic_pdc_sex, pVal_female < 0.1 | pVal_male < 0.1),
                        digits = 3, booktabs = TRUE, row.names = FALSE), font_size = 7)
```

var	C_female	P_female	pVal_female	C_male	P_male	pVal_male
Total SCFAs	3670.79 ± 2214.48	3301.34 ± 2513.4	0.335	4787.37 ± 3056.69	3277.41 ± 2314.98	0.052
Acetic acid	2012.88 ± 1374.1	1965.12 ± 1849.39	0.487	2796.4 ± 1936.46	1914.84 ± 1523.55	0.064
Propionic acid	781.83 ± 480.18	640.39 ± 283.66	0.388	902.75 ± 541.49	647.95 ± 407.75	0.057
Butyric acid	658.33 ± 661.11	436.15 ± 461.58	0.100	819.95 ± 604.11	414.26 ± 356.8	0.003
Stool calprotectin	665.37 ± 772.91	1502.36 ± 1696.54	0.019	1102.49 ± 1178.53	1497.26 ± 1571.93	0.489
Stool IFNg	6.41 ± 4.01	4.85 ± 3.83	0.078	6.27 ± 6.83	6.42 ± 5.49	0.282
Stool IL10	0.76 ± 0.49	0.6 ± 0.61	0.090	0.71 ± 0.94	0.73 ± 0.77	0.353
Stool IL12p70	1.44 ± 0.86	1.02 ± 0.89	0.040	1.27 ± 1.45	1.22 ± 1.19	0.558
Stool IL13	6.73 ± 3.63	5.15 ± 3.86	0.065	5.76 ± 4.98	6.24 ± 4.11	0.233
Stool IL2	2.09 ± 1.23	1.52 ± 1.24	0.070	1.85 ± 2.09	2.24 ± 2.27	0.312
Stool IL4	0.29 ± 0.18	0.23 ± 0.23	0.078	0.33 ± 0.45	0.26 ± 0.25	0.591
Stool IL6	1.35 ± 0.9	0.94 ± 0.8	0.047	1.23 ± 1.44	1.34 ± 1.22	0.304
Plasma IL13	7.03 ± 6.64	6.53 ± 6.78	0.796	3.55 ± 3.65	6.44 ± 5.3	0.006
Plasma IL4	0.08 ± 0.06	0.07 ± 0.06	0.498	0.04 ± 0.03	0.07 ± 0.06	0.057
Plasma CXCL8	5.25 ± 1.37	4.67 ± 1.94	0.017	5.68 ± 2.29	4.95 ± 2.13	0.145

```
# Combine with the results for full data
table_basic_pdc_final <- cbind(table_basic_pdc, table_basic_pdc_sex[, 2:7])

# Export
write.csv(table_basic_pdc_final, "Outputs/tableS1.csv", row.names = FALSE)
```

1.3.0.4 Fig 1

Boxplots for the variables that differ significantly in the full data and one of the subsets:

```
# Data frame for figure
basic_pdc_fig_vars <- row.names(subset(table_basic_pdc_final,
                                       as.numeric(as.character(pVal)) < 0.05 &
                                       (pVal_female < 0.05 | pVal_male < 0.05)))
basic_pdc_fig_df <- melt(pdcy_meta[, c("sex", "Group", basic_pdc_fig_vars)])

# Better scales
basic_pdc_fig_df[basic_pdc_fig_df$variable == "butyric_acid", "value"] <-
  basic_pdc_fig_df[basic_pdc_fig_df$variable == "butyric_acid", "value"]/1000
basic_pdc_fig_df[basic_pdc_fig_df$variable == "Scalprotectin", "value"] <-
  basic_pdc_fig_df[basic_pdc_fig_df$variable == "Scalprotectin", "value"]/1000

# Data frame for p-value annotations
pdc_sig_pstars <- subset(data.frame(Var = rownames(table_basic_pdc_final),
                                   pVal_F = table_basic_pdc_final$pVal_female,
                                   pVal_M = table_basic_pdc_final$pVal_male),
                        Var %in% basic_pdc_fig_vars)
pdc_sig_pstars <- subset(melt(pdc_sig_pstars), value < 0.05)
pdc_sig_pstars$sex <- sub(".*_", "", pdc_sig_pstars$variable)
pdc_sig_pstars$stars <- stars.pval(pdc_sig_pstars$value)
pdc_sig_pstars$y <- c(6.2, 13.5, 2.8)
```

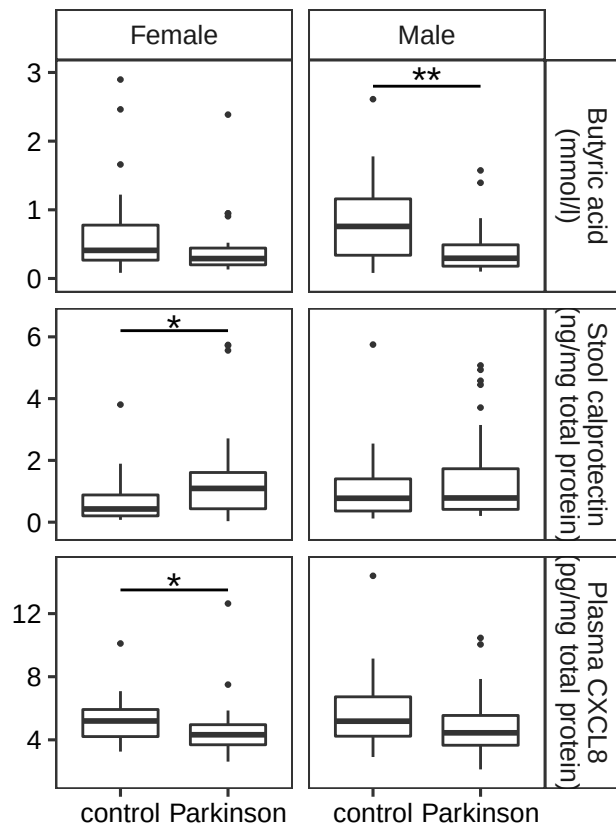
```

pdc_sig_pstars$ytext <- c(6.3, 13.8, 2.9)
colnames(pdc_sig_pstars)[1:2] <- c("variable", "pVariable")

# Control/PD labels
pdc_labels <- setNames(c("control", "Parkinson"), c("C", "P"))

# Plot
pdc_sig_plots <- ggplot(basic_pdc_fig_df, aes(x = Group, y = value)) +
  geom_boxplot(width = 0.7, outlier.size = 0.5) +
  theme_bw(base_size = 12) +
  scale_x_discrete(labels = pdc_labels) +
  scale_y_continuous(expand = expansion(mult = 0.1)) +
  facet_grid(variable ~ sex, scales = "free", labeller = labeller(
    variable = setNames(c("Butyric acid\n(mmol/l)",
                          "Plasma CXCL8\n(pg/mg total protein)",
                          "Stool calprotectin\n(ng/mg total protein)"),
                        c("butyric_acid", "pCXCL8", "Scalprotectin")),
    sex = setNames(c("Female", "Male"), c("F", "M")))) +
  geom_text(aes(x = 1.5, y = ytext, label = stars),
            pdc_sig_pstars, inherit.aes = FALSE, size = 5) +
  geom_segment(aes(x = 1, xend = 2, y = y, yend = y),
              pdc_sig_pstars, inherit.aes = FALSE) +
  theme(panel.grid = element_blank(),
        axis.text = element_text(color = "black"),
        axis.title = element_blank(),
        strip.background = element_rect(fill = "white"))
pdc_sig_plots

```



```
# Export
ggsave(pdc_sig_plots, filename = "Outputs/fig1_pdc_sigs.tiff",
       width = 90, height = 120, units = "mm", dpi = 600,
       type = "cairo", compression = "lzw")
```

2 Comparisons without microbiota data

2.1 Intercorrelations of inflammatory markers and SCFAs

2.1.0.1 Fig S1

Run Pearson correlations for pairs of inflammatory markers as well as the SCFAs to test whether they are intercorrelated. Do this for full data as well as control-only and PD-only subsets. Make combined plots of the results.

```
# Full data
ims_cors <- rcorr(as.matrix(
  pdcy_meta[, c(scfa_vars, inf_vars_p, inf_vars_s)]), type = "pearson")

# PD-only data
ims_cors_pd <- rcorr(as.matrix(
  subset(pdcy_meta, Group == "P")[, c(scfa_vars, inf_vars_p, inf_vars_s)]),
  type = "pearson")

# C-only data
ims_cors_c <- rcorr(as.matrix(
  subset(pdcy_meta, Group == "C")[, c(scfa_vars, inf_vars_p, inf_vars_s)]),
  type = "pearson")

custom_corrplot <- function(cmat, pmat){
  ggcorrplot(cmat, type = "upper", p.mat = pmat,
             tl.cex = 6, colors = c("black", "white", "firebrick4"),
             insig = "pch", pch = 32) +
  geom_point(aes(shape = factor(pvalue < 0.05)), size = 0.75) +
  scale_shape_manual(values = c(32, 8), name = "p < 0.05",
                             labels = c("no", "yes")) +
  scale_x_discrete(position = "top") +
  geom_segment(inherit.aes = FALSE,
              data = data.frame(
                x = c(0, 0, 7.5, 18.5),
                y = c(6.5, 17.5, 30.5, 30.5),
                xend = c(7.5, 18.5, 7.5, 18.5),
                yend = c(6.5, 17.5, 6.5, 17.5), size = 1),
              aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_text(inherit.aes = FALSE,
           data = data.frame(
             x = c(4, 10, 22),
             y = c(1, 7, 19),
             label = c("SCFAs", "Plasma markers", "Stool markers")),
           aes(x = x, y = y, label = label),
           hjust = 0, angle = 45) +
  theme(axis.text.x = element_text(hjust = 0, vjust = 0.5, color = "black",
                                    angle = 45),
        axis.text.y = element_text(color = "black"),
```

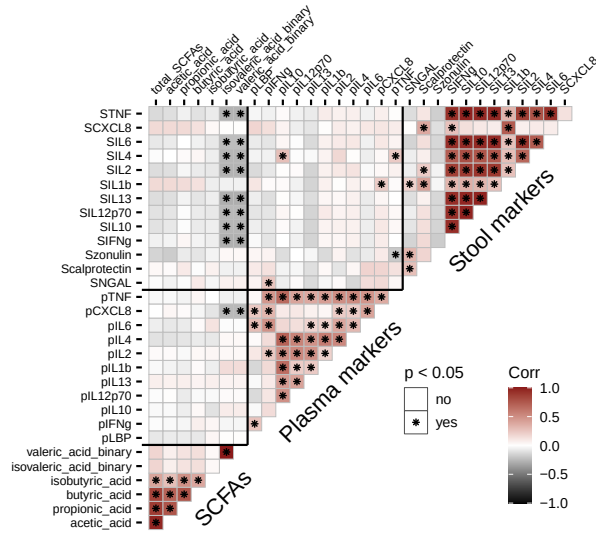
```

    axis.ticks = element_line(color = "black"),
    panel.grid = element_blank(),
    legend.key = element_rect(color = "gray30", size = 0.2),
    legend.key.size = unit(0.4, "cm"),
    legend.box.margin = margin(l = -3.5, t = 4, unit = "cm"),
    legend.box = "horizontal",
    legend.text = element_text(size = 7),
    legend.title = element_text(size = 8))
}

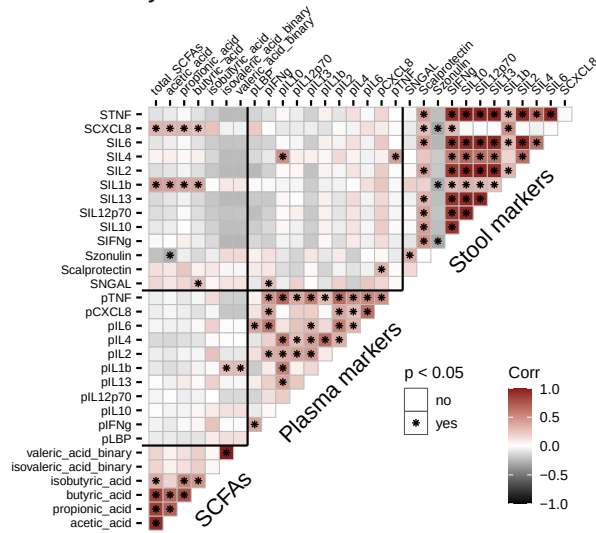
marker_scfa_cor_plots <- arrangeGrob(
  custom_corrplot(ims_cors$r, ims_cors$P) +
    ggtitle("A. Full data") +
    theme(plot.title = element_text(vjust = -1, hjust = -0.55)),
  custom_corrplot(ims_cors_c$r, ims_cors_c$P) +
    ggtitle("B. Control only") +
    theme(plot.title = element_text(vjust = -1, hjust = -0.6)),
  custom_corrplot(ims_cors_pd$r, ims_cors_pd$P) +
    ggtitle("C. PD only") +
    theme(plot.title = element_text(vjust = -1, hjust = -0.5)),
  ncol = 1)
grid.arrange(marker_scfa_cor_plots)

```

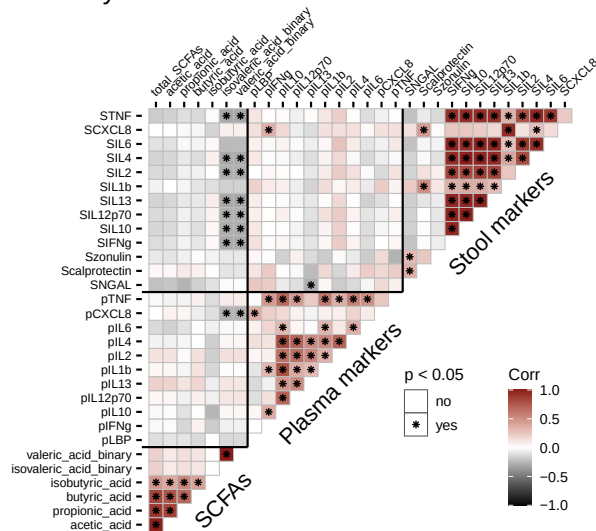
A. Full data



B. Control only



C. PD only



```
ggsave(marker_scfa_cor_plots,
       filename = "Outputs/figS1_scfa_marker_cors.pdf",
       width = 4.5, height = 12, units = "in")
```

2.2 PCA with inflammatory markers

Since many of the markers were intercorrelated, use Principal Component Analysis to condense those variables (goal: decrease the number of variables in comparisons and avoid collinearity issues).

2.2.1 Stool

For stool, leave out NGAL, calprotectin, zonulin, IL1b and CXCL8 (which were less strongly intercorrelated with other variables) and include the rest of the variables.

```
infs_PCA_vars <- inf_vars_s[!(inf_vars_s %in% c("SNGAL", "Scalprotectin", "Szonulin",
                                              "SIL1b", "SCXCL8"))]

# Scale the values
infs_scaled <- scale(pdcy_meta[, infs_PCA_vars])
infs_scaled <- as.data.frame(infs_scaled[1:nrow(infs_scaled), ])

# Calculate PCA
infs_PCA <- prcomp(as.formula(paste("~", paste(infs_PCA_vars, collapse = " + "))),
                  data = infs_scaled, na.action = na.omit)
```

The proportion of variance explained by the PCA axes as a plot:

```
fviz_screplot(infs_PCA, ncp = 15) +
  theme_bw(base_size = 9) +
  ylab("Percentage of\nexplained variances") +
  theme(panel.grid.major.x = element_blank())
```

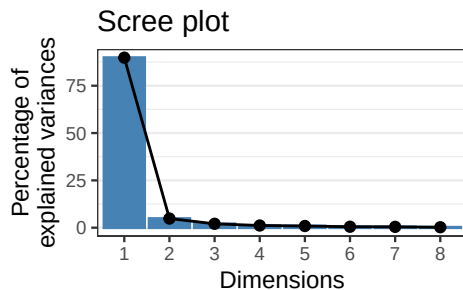


Table of metrics for all principal components:

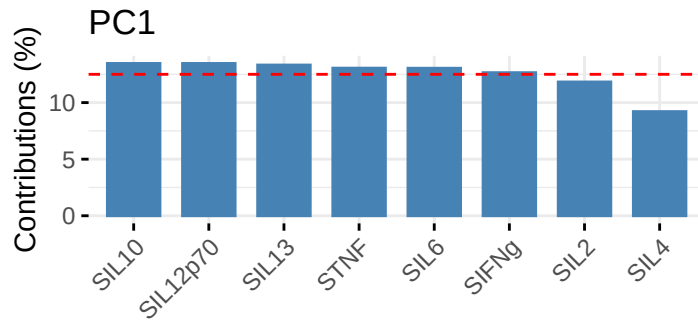
```
kable(summary(infs_PCA)$importance, digits = 3, booktabs = TRUE)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Standard deviation	2.681	0.622	0.406	0.304	0.268	0.203	0.191	0.140
Proportion of Variance	0.898	0.048	0.021	0.012	0.009	0.005	0.005	0.002
Cumulative Proportion	0.898	0.947	0.967	0.979	0.988	0.993	0.998	1.000

The first PC captured almost 90% of all variance. It's probably enough to just keep the first axis for further analyses.

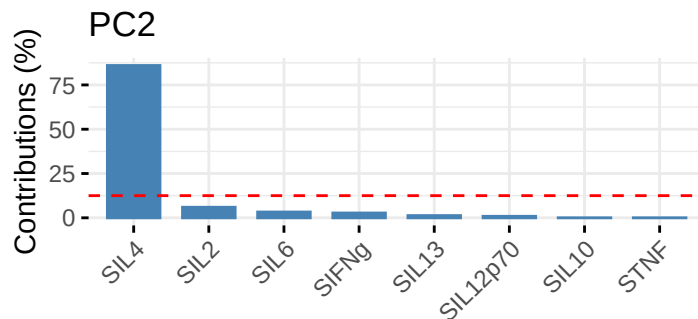
Plots of the variables contributing to the first three principal components:

```
fviz_contrib(infs_PCA, choice = "var", axes = 1) + ggtitle("PC1")
```



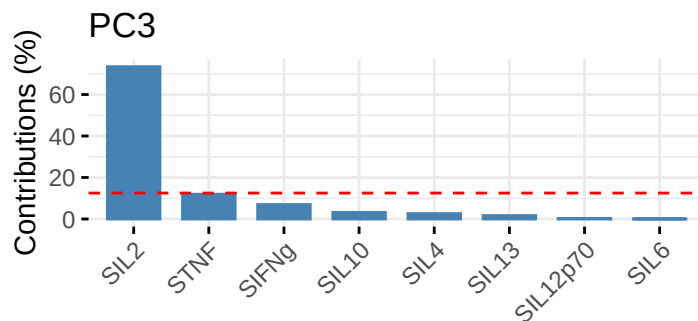
PC1 had significant contributions from most ILs, TNF and IFNg, with IL2 and IL4 falling beneath the reference line.

```
fviz_contrib(infs_PCA, choice = "var", axes = 2) + ggtitle("PC2")
```



PC2 corresponded to IL4.

```
fviz_contrib(infs_PCA, choice = "var", axes = 3) + ggtitle("PC3")
```



PC3 corresponded to IL2.

Plots of the first three axes, with the PD and control subjects marked:

```
library("ggfortify")
```

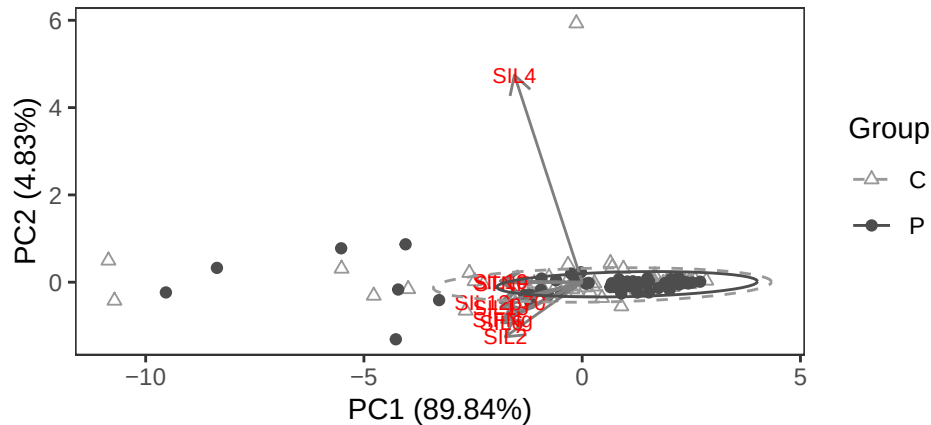
```
pcaPlotPD <- function(pcdat, df, x, y){
  autoplot(pcdat, data = df, x = x, y = y, colour = "Group", shape = "Group",
    loadings = TRUE, loadings.colour = "gray50", loadings.label = TRUE,
    loadings.label.size = 2.75, scale = 0) +
  theme_bw() +
  coord_fixed() +
  stat_ellipse(aes(group = Group, color = Group, lty = Group), level = 0.9) +
  scale_color_manual(values = c("gray60", "gray30"), name = "Group") +
```

```

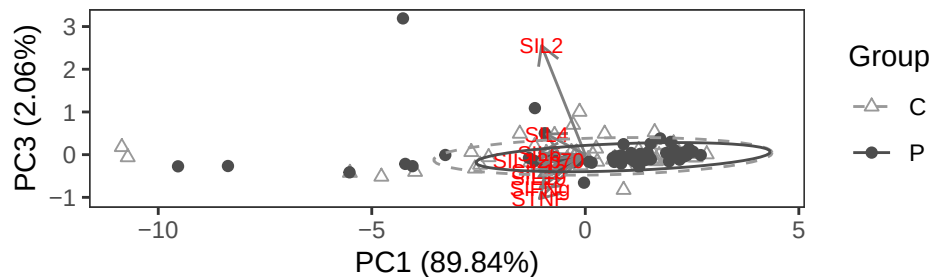
scale_shape_manual(values = c(2, 19), name = "Group") +
scale_linetype_manual(values = c(2, 1), name = "Group") +
theme(panel.grid = element_blank())
}

```

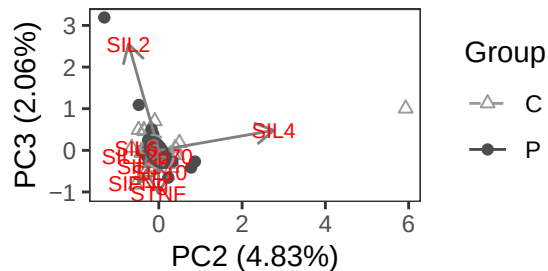
```
pcaPlotPD(infs_PCA, pdcy_meta[rownames(infs_PCA$x), ], 1, 2)
```



```
pcaPlotPD(infs_PCA, pdcy_meta[rownames(infs_PCA$x), ], 1, 3)
```



```
pcaPlotPD(infs_PCA, pdcy_meta[rownames(infs_PCA$x), ], 2, 3)
```



Export PC1 and add to the metadata as "StoolPC" - a summary variable representing IFNg, IL10, IL12p70, IL13, IL6 and TNF. Keep IL4 and IL2 as individual variables.

```

infs_PC1 <- data.frame(infs_PCA$x[, 1])

# Export
write.csv(infs_PC1, "Outputs/infs_pca.csv")

# Add to full data
pdcy_meta$StoolPC <- infs_PC1[rownames(pdcy_meta), ]

```



```
# New list of variables of interest for stool (stool markers + PC1)
infs_vars <- c("SNGAL", "Scalprotectin", "Szonulin", "SCXCL8", "SIL1b",
               "SIL4", "SIL2", "StoolPC")
```

2.2.2 Plasma

For plasma marker PCA, leave out LBP, IFNg, IL6 and CXCL8, which were not as strongly intercorrelated as the rest of the variables, and run the analysis with the remaining variables.

```
infp_PCA_vars <- inf_vars_p[!(inf_vars_p %in% c("pLBP", "pIFNg", "pCXCL8", "pIL6"))]

# Scale the values
infp_scaled <- scale(pdcy_meta[, infp_PCA_vars])
infp_scaled <- as.data.frame(infp_scaled[1:nrow(infp_scaled), ])

# Calculate PCA
infp_PCA <- prcomp(as.formula(paste("~", paste(infp_PCA_vars, collapse = " + "))),
                  data = infp_scaled, na.action = na.omit)
```

The proportion of variance explained by the resulting PCA axes as a plot:

```
fviz_screplot(infp_PCA, ncp = 15) +
  theme_bw(base_size = 9) +
  ylab("Percentage of\nexplained variances") +
  theme(panel.grid.major.x = element_blank())
```

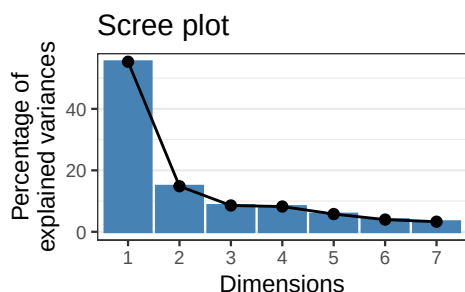


Table of metrics for all principal components:

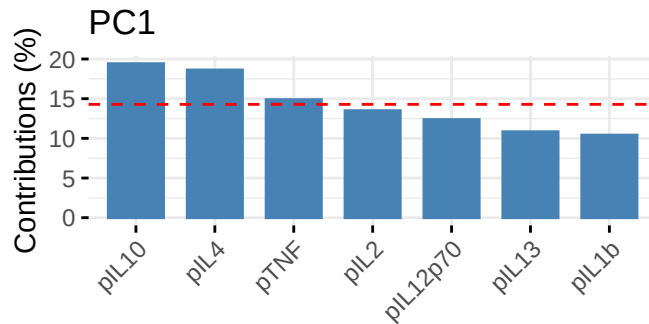
```
kable(summary(infp_PCA)$importance[1:3, ], digits = 3, booktabs = TRUE)
```

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	1.968	1.019	0.774	0.759	0.636	0.527	0.480
Proportion of Variance	0.553	0.148	0.086	0.082	0.058	0.040	0.033
Cumulative Proportion	0.553	0.702	0.787	0.870	0.927	0.967	1.000

The situation was not as clear as for the stool variables, but PC1 did explain 55% of all variance. One could consider keeping either just PC1, or the first two PCs.

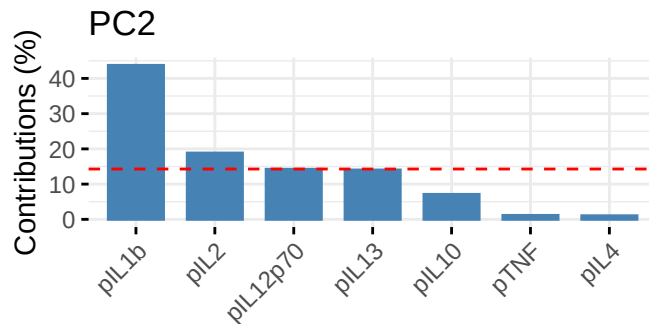
Plots of the variables contributing to the first four principal components:

```
fviz_contrib(infp_PCA, choice = "var", axes = 1) + ggtitle("PC1")
```



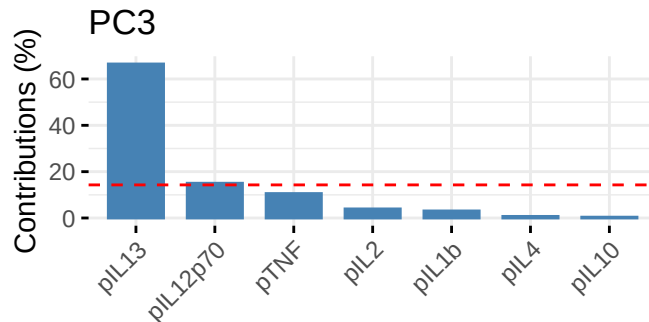
PC1 had contributions mostly from IL10, IL4 and TNF.

```
fviz_contrib(infp_PCA, choice = "var", axes = 2) + ggtitle("PC2")
```



PC2 was mainly IL1b, with minor contributions from IL2, IL12p70 and IL13.

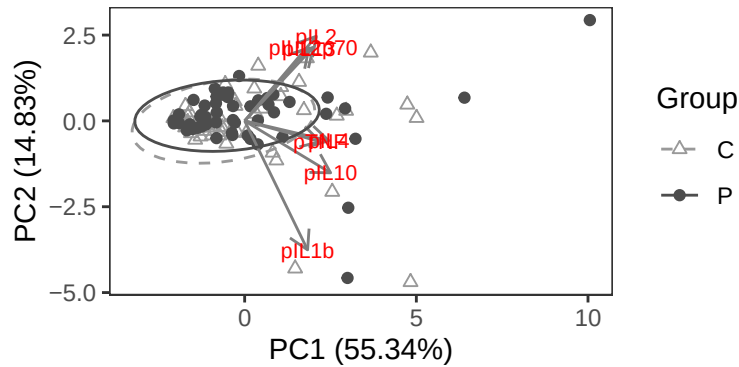
```
fviz_contrib(infp_PCA, choice = "var", axes = 3) + ggtitle("PC3")
```



PC3 was mostly IL13 and a minor contribution from IL12p70.

Plot of PC1 and PC2, with the PD and control subjects marked:

```
pcaPlotPD(infp_PCA, pdcy_meta[rownames(infp_PCA$x), ], 1, 2)
```



Export the first two axes for downstream use:

```
infp_PCs <- data.frame(infp_PCA$x[, 1:2])

# Export
write.csv(infp_PCs, "Outputs/infp_pca.csv")

# Add to data
infp_PCs <- infp_PCs[rownames(pdcy_meta), ]
pdcy_meta$pPC1 <- infp_PCs$PC1
pdcy_meta$pPC2 <- infp_PCs$PC2

# New list of variables of interest for plasma
infp_vars <- c("pLBP", "pIFNg", "pCXCL8", "pIL6", "pPC1", "pPC2")

# Final list of marker/SCFA variables of interest
inf_scfa_vars <- c(scfa_vars, inf_vars, infp_vars)

# Full list of all marker/SCFA variables (including redundant ones)
inf_scfa_all_vars <- unique(c(inf_scfa_vars, inf_vars_s, inf_vars_p,
                             "valeric_acid", "isovaleric_acid"))
```

2.3 Correlations between clinical variables

To test for correlations between variables, run a correlation matrix for the inflammatory markers, SCFAs, and potential confounding variables that are either numeric or binary factors. Run tests for full data and control-only and PD-only subsets, with the PD-only comparisons including additional variables that are only relevant to PD patients.

```
# Numeric variables
cor_vars <- names(which(sapply(pdcy_meta, is.numeric)))

# Binary variables
cor_vars_b <- names(which(sapply(pdcy_meta, function(x) length(levels(x)) == 2)))
# Make binary numeric variables out of the factor ones
cor_vars_b_num <- sapply(pdcy_meta[, cor_vars_b], as.numeric)

# Put together data matrix
cor_data <- cbind(as.matrix(pdcy_meta[, cor_vars]), cor_vars_b_num)
cor_data <- cor_data[, order(colnames(cor_data))]

# Full list of all confounding variables
conf_vars <- colnames(cor_data)
```

```

# Exclude some non-relevant or too uncommon variables +
# the PD vs control, SCFA and marker variables
conf_vars <- conf_vars[!(conf_vars %in% c("Group", "LED", "DBS",
                                           "history_gyn_prolapse",
                                           "history_gyn_prolapse_repair",
                                           "functional_constipation_criteria_fulfilled",
                                           "history_endometriosis",
                                           "Rome_III_IBS_possible",
                                           "functional_constipation_possible",
                                           inf_scfa_all_vars))]

# Trim PD-only relevant confounders (hand-picked) from full data confounders
conf_vars_no_pd <- conf_vars[!(conf_vars %in% unique(c(
  grep("onset", conf_vars, value = TRUE),
  grep("poletti", conf_vars, value = TRUE),
  grep("jankovic", conf_vars, value = TRUE),
  grep("dopa", conf_vars, value = TRUE, ignore.case = TRUE),
  grep("mg", conf_vars, value = TRUE),
  "meds_COMT_inhibitor", "meds_MAO_inhibitor", "Hoehn_and_Yahr_OFF",
  grep("UPDRS", conf_vars, value = TRUE)))))]

# Run correlations for full data (trimmed list of variables)
meta_cors <- rcorr(cor_data[, c(conf_vars_no_pd, "Group", inf_scfa_vars)],
                  type = "pearson")

# Run correlations for C-only (trimmed list of variables)
meta_cors_c <- rcorr(cor_data[rownames(subset(pdcy_meta, Group == "C")),
                              c(conf_vars_no_pd, inf_scfa_vars)],
                    type = "pearson")

# Run correlations for PD-only (all variables)
meta_cors_pd <- rcorr(cor_data[rownames(subset(pdcy_meta, Group == "P")),
                              c(conf_vars, inf_scfa_vars)],
                    type = "pearson")

```

2.3.1 SCFAs, markers vs other variables

Look for significant correlations between SCFAs, immune markers and clinical variables.

2.3.1.1 Fig 2

Plot full data, control-only and PD-only comparisons side by side, excluding the clinical variables that have no correlations with $p < 0.05$ for any comparison in any subset.

```

# Data frame of p-values
meta_cors_P <- rbind(
  melt(data.frame(meta_cors$P[conf_vars_no_pd, c("Group", inf_scfa_vars)],
                  Confounder = conf_vars_no_pd,
                  Set = "All subjects")),
  melt(data.frame(meta_cors_c$P[conf_vars_no_pd, inf_scfa_vars],
                  Confounder = conf_vars_no_pd,
                  Set = "Control subjects")),
  melt(data.frame(meta_cors_pd$P[conf_vars, inf_scfa_vars],
                  Confounder = conf_vars,

```

```

        Set = "PD patients"))))
# Add NAs for the variables not in control/full data
# (purely for plotting purposes)
pd_only_vars <- setdiff(conf_vars, conf_vars_no_pd)
meta_cors_P <- rbind(meta_cors_P,
  expand.grid(
    Confounder = pd_only_vars,
    Set = c("All subjects", "Control subjects"),
    variable = inf_scfa_vars,
    value = 1),
  data.frame(
    Confounder = pd_only_vars,
    Set = "All subjects",
    variable = "Group",
    value = 1)
)

# Data frame of r values
meta_cors_r <- rbind(
  melt(data.frame(meta_cors$r[conf_vars_no_pd, c("Group", inf_scfa_vars)],
    Confounder = conf_vars_no_pd,
    Set = "All subjects")),
  melt(data.frame(meta_cors_c$r[conf_vars_no_pd, inf_scfa_vars],
    Confounder = conf_vars_no_pd,
    Set = "Control subjects")),
  melt(data.frame(meta_cors_pd$r[conf_vars, inf_scfa_vars],
    Confounder = conf_vars,
    Set = "PD patients")))
# Add NAs for the variables not in control/full data
meta_cors_r <- rbind(meta_cors_r,
  expand.grid(
    Confounder = pd_only_vars,
    Set = c("All subjects", "Control subjects"),
    variable = inf_scfa_vars,
    value = NA),
  data.frame(
    Confounder = pd_only_vars,
    Set = "All subjects",
    variable = "Group",
    value = NA))

# Fix labels

# Main variables
inf_scfa_vars2 <- sub(" binary", "",
  sub(" tool", " ",
    varLabelFix(inf_scfa_vars)))
names(inf_scfa_vars2) <- inf_scfa_vars
inf_scfa_vars2 <- c('Group' = "PD vs control", inf_scfa_vars2)

# Confounders
meta_cors_P$Confounder_simple <- meta_cors_P$Confounder
meta_cors_P$Confounder <-

```

```

sub("dopa$", "levodopa",
  sub("nms", "NMS",
    sub("constip defec.*", "9-15 sum score",
      sub("janko", "Janko",
        sub("polet", "Polet",
          sub("pigd", "PIGD", ignore.case = TRUE,
            sub("ar", "AR", ignore.case = TRUE,
              sub("Meds", "Medication",
                sub("ca ", "Ca ",
                  sub(" OFF", "",
                    sub("mg", "(mg)",
                      gsub("_", " ",
                        capitalize(
                          as.character(
                            meta_cors_P$Confounder))))))))))))))
meta_cors_P$Confounder <- factor(meta_cors_P$Confounder,
  levels = unique(meta_cors_P$Confounder))
# Better label for the tobacco variable
levels(meta_cors_P$Confounder)[grep("Tobacco", levels(meta_cors_P$Confounder))] <-
  "Smoking (>100 cigarettes in life)"

# Same fixes to the df with r values
meta_cors_r$Confounder <- meta_cors_P$Confounder

# Separate subsets per type of variable

# SCFAs
scfa_cor_sigs <- names(which(table(
  subset(meta_cors_P, variable %in% scfa_vars)$Confounder,
  subset(meta_cors_P, variable %in% scfa_vars)$value < 0.05)[,2]>0))
meta_cors_sc_P <- subset(meta_cors_P, Confounder %in% scfa_cor_sigs &
  variable %in% c("Group", scfa_vars))
meta_cors_sc_P$value <- meta_cors_sc_P$value < 0.05
meta_cors_sc_r <- subset(meta_cors_r, Confounder %in% scfa_cor_sigs &
  variable %in% c("Group", scfa_vars))

# Plasma markers
p_cor_sigs <- names(which(table(
  subset(meta_cors_P, variable %in% infp_vars)$Confounder,
  subset(meta_cors_P, variable %in% infp_vars)$value < 0.05)[,2]>0))
meta_cors_pl_P <- subset(meta_cors_P, Confounder %in% p_cor_sigs &
  variable %in% c("Group", infp_vars))
meta_cors_pl_P$value <- meta_cors_pl_P$value < 0.05
meta_cors_pl_r <- subset(meta_cors_r, Confounder %in% p_cor_sigs &
  variable %in% c("Group", infp_vars))

# Stool markers
s_cor_sigs <- names(which(table(
  subset(meta_cors_P, variable %in% infs_vars)$Confounder,
  subset(meta_cors_P, variable %in% infs_vars)$value < 0.05)[,2]>0))
meta_cors_st_P <- subset(meta_cors_P, Confounder %in% s_cor_sigs &
  variable %in% c("Group", infs_vars))
meta_cors_st_P$value <- meta_cors_st_P$value < 0.05

```

```

meta_cors_st_r <- subset(meta_cors_r, Confounder %in% s_cor_sigs &
                        variable %in% c("Group", infs_vars))

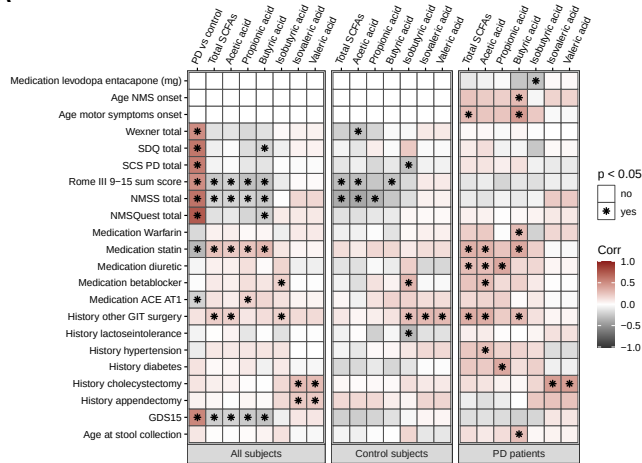
# Plotting function
custom_corrplot2 <- function(cmat, pmat){
  ggplot(cmat, aes(x = variable, y = Confounder,
                  fill = value)) +

  geom_tile(color = "gray30") +
  theme_bw(base_size = 6) +
  facet_grid(~Set, space = "free", scales = "free", switch = "x") +
  scale_x_discrete(position = "top", labels = inf_scfa_vars2) +
  xlab(NULL) + ylab(NULL) +
  scale_fill_gradient2(low = "gray20", mid = "white",
                      high = "firebrick4", name = "Corr",
                      limits = c(-1, 1), na.value = "white") +
  geom_point(data = pmat,
            aes(x = variable, y = Confounder,
                shape = factor(value)),
            inherit.aes = FALSE, size = 0.75, color = "black") +
  scale_shape_manual(values = c(32, 8), name = "p < 0.05",
                      labels = c("no", "yes")) +
  theme(axis.text = element_text(color = "black"),
        axis.text.x.top = element_text(angle = 65,
                                         hjust = 0, vjust = 0),
        axis.ticks = element_line(color = "black"),
        legend.key.size = unit(0.3, "cm"),
        plot.margin = margin(l = 0, r = 4, b = 2, unit = "mm"),
        legend.key = element_rect(color = "gray30", size = 0.2),
        panel.grid = element_blank(),
        aspect.ratio = 1)
}

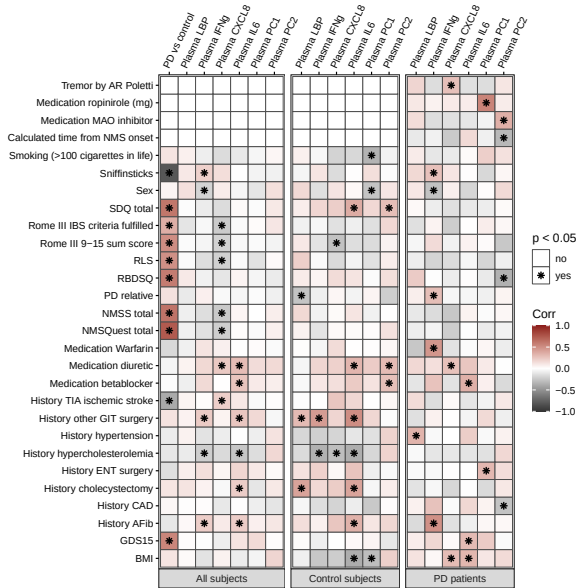
# Make plots
# (title positions adjusted manually by trial and error)
clin_cors_plots <- grid.arrange(
  custom_corrplot2(meta_cors_sc_r, meta_cors_sc_P) +
  ggtitle("A") +
  theme(plot.title = element_text(face = "bold", size = 11,
                                   vjust = -1,
                                   hjust = -0.51)),
  custom_corrplot2(meta_cors_pl_r, meta_cors_pl_P) +
  ggtitle("B") +
  theme(plot.title = element_text(face = "bold", size = 11,
                                   vjust = -1,
                                   hjust = -0.52)),
  custom_corrplot2(meta_cors_st_r, meta_cors_st_P) +
  ggtitle("C") +
  theme(plot.title = element_text(face = "bold", size = 11,
                                   vjust = -1,
                                   hjust = -0.3)),
  ncol = 1, heights = c(0.8, 1, 1)
)

```

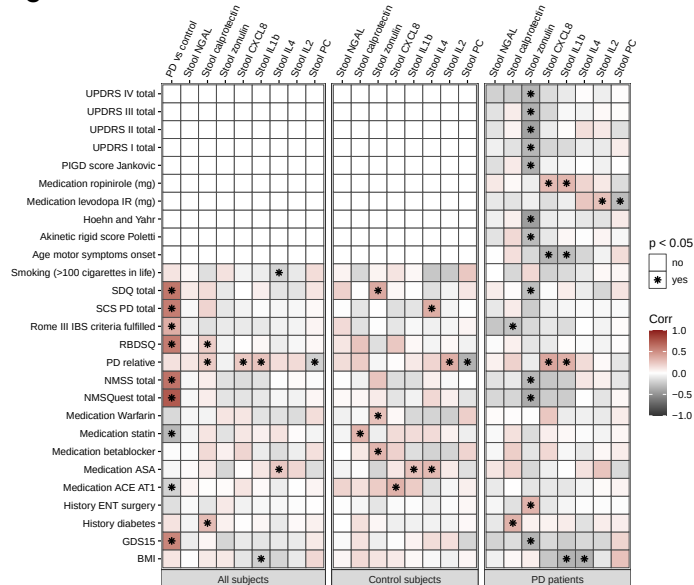
A



B



C




```
# Export
ggsave(clin_cors_plots, filename = "Outputs/fig2_clin_cors.tiff",
       width = maxwd/2, height = maxhi, units = "mm", dpi = 600,
       type = "cairo", compression = "lzw")
```

2.3.1.2 Fig S2

Scatterplots for the most interesting pairs of variables:

```
# Basic scatterplot function
basic_scatter <- function(df, xvar, yvar){
  ggplot(df, aes_string(x = xvar, y = yvar,
                        color = "Group", shape = "Group")) +
    geom_point(size = 0.5) +
    scale_color_manual(values = c("gray50", "black"),
                      labels = c("control", "Parkinson")) +
    scale_shape_manual(values = c(16, 17),
                      labels = c("control", "Parkinson")) +
    theme_bw(base_size = 4.5) +
    theme(panel.grid = element_blank(),
          axis.text = element_text(color = "black"),
          aspect.ratio = 1)
}

# Basic boxplot for categorical variables
basic_box <- function(df, xvar, yvar){
  ggplot(df, aes_string(x = xvar, y = yvar, color = "Group")) +
    geom_boxplot(outlier.size = 0.5, lwd = 0.3) +
    scale_x_discrete(labels = c("no", "yes")) +
    scale_color_manual(values = c("gray50", "black"),
                      labels = c("control", "Parkinson")) +
    theme_bw(base_size = 4.5) +
    theme(panel.grid = element_blank(),
          axis.text = element_text(color = "black"),
          aspect.ratio = 1)
}

# Basic scatterplot function (PD only)
basic_scatter_pd <- function(df, xvar, yvar){
  df <- subset(df, Group == "P")
  ggplot(df, aes_string(x = xvar, y = yvar)) +
    geom_point(shape = 17, size = 0.5) +
    theme_bw(base_size = 4.5) +
    theme(panel.grid = element_blank(),
          axis.text = element_text(color = "black"),
          aspect.ratio = 1)
}

# Basic boxplot function (PD only)
basic_box_pd <- function(df, xvar, yvar){
  df <- subset(df, Group == "P")
  ggplot(df, aes_string(x = xvar, y = yvar)) +
    geom_boxplot(outlier.size = 0.5, lwd = 0.3) +
    scale_x_discrete(labels = c("no", "yes")) +
```

```

    theme_bw(base_size = 4.5) +
    theme(panel.grid = element_blank(),
          axis.text = element_text(color = "black"),
          aspect.ratio = 1)
}

# Get all pairs of significant correlations from full data and PD-only comparisons
sig_cors <- subset(meta_cors_P, value < 0.05 & variable != "Group" &
                  Set != "Control subjects")

# Rearrange
sig_cors <- dcast(sig_cors, variable + Confounder + Confounder_simple ~ Set)
# Trim to PD-related confounders of interest
sig_cors <- subset(sig_cors,
                  Confounder_simple %in% c("age_NMS_onset", "NMSS_total",
                                           "Rome_III_constip_defec_sumscore_9.15",
                                           "Rome_III_IBS_criteria_fulfilled",
                                           "GDS15",
                                           "UPDRS_III_total_OFF", "Hoehn_and_Yahr_OFF",
                                           "SDQ_total", "age_motor_symptoms_onset",
                                           "meds_statin", "history_TIA_ischemic_stroke"))

# Exclude propionic and acetic acid, since these
# tend to be similar to the total SCFAs and butyric acid
sig_cors <- subset(sig_cors, variable != "propionic_acid" &
                  variable != "acetic_acid")

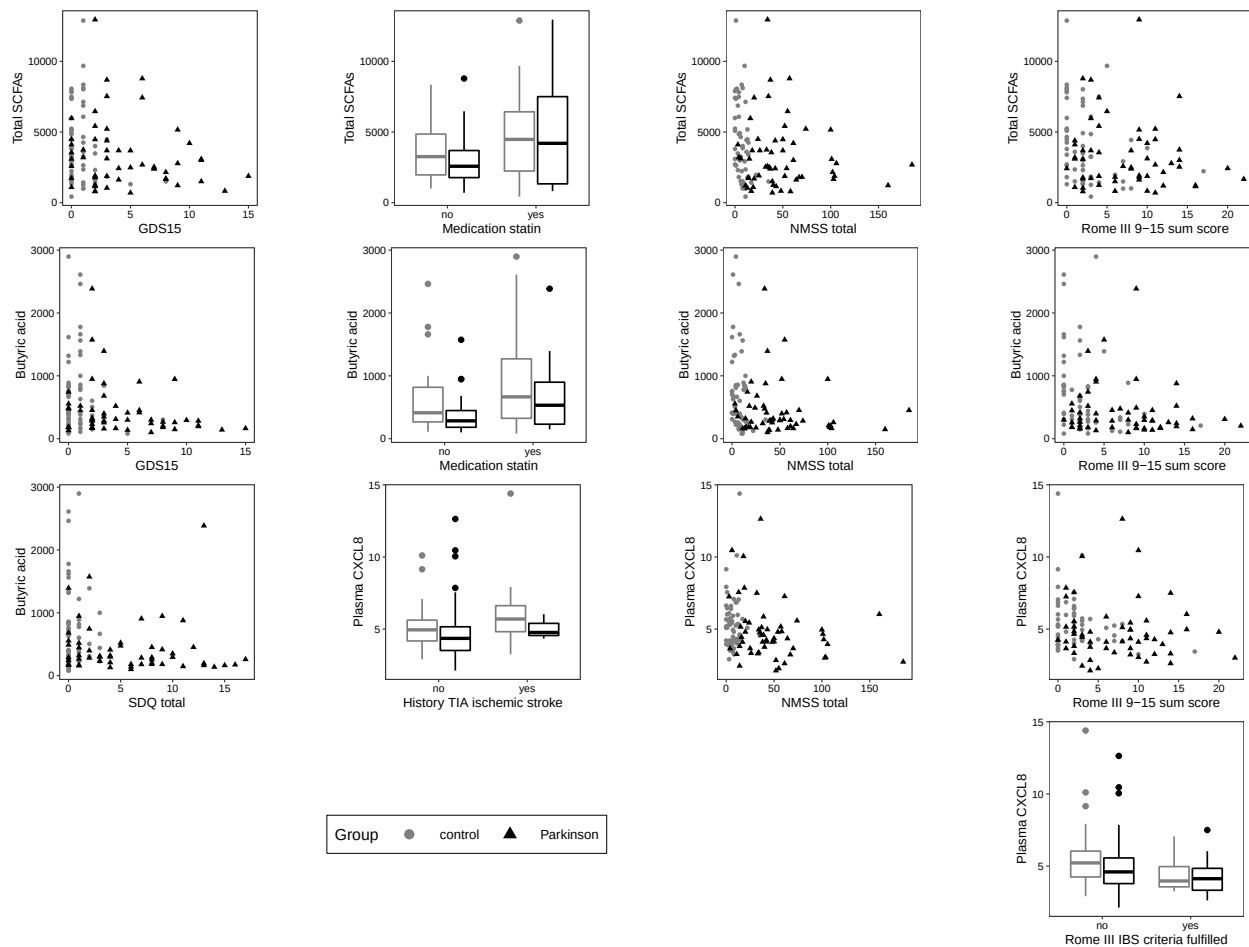
# Plot variables significant for full data
conf_cors_full <- apply(subset(sig_cors, `All subjects` < 0.05), 1,
                      function(x){
                        if(is.numeric(pdcy_meta[, x["Confounder_simple"]])) {
                          basic_scatter(pdcy_meta, x["Confounder_simple"],
                                         x["variable"]) +
                          xlab(x["Confounder"]) +
                          ylab(inf_scfa_vars2[x["variable"]]) +
                          theme(legend.position = "none")
                        } else {
                          basic_box(
                            subset(pdcy_meta, !is.na(
                              pdcy_meta[, as.character(
                                as.matrix(x["Confounder_simple"]))))),
                            x["Confounder_simple"],
                            x["variable"]) +
                          xlab(x["Confounder"]) +
                          ylab(inf_scfa_vars2[x["variable"]]) +
                          theme(legend.position = "none")
                        }
                      })

# Legend-grabbing function
gLegend <- function(a.gplot){
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  legend
}

```

```
# Get legend for the full-data plots
clin_plot_legend <- gLegend(
  basic_scatter(pdcy_meta, "GDS15", "total_SCFAs") +
  geom_point(size = 1.5) +
  theme_bw(base_size = 6) +
  theme(legend.background = element_rect(color = "black", size = 0.1),
        legend.key.height = unit(3, "mm"),
        legend.position = "bottom")
)

grid.arrange(
  do.call("arrangeGrob",
    c(conf_cors_full[1:(length(conf_cors_full)-1)], ncol = 4)),
  arrangeGrob(clin_plot_legend, conf_cors_full[[length(conf_cors_full)]],
    nrow = 1, widths = c(3, 1)),
  nrow = 2, heights = c(3, 1))
```



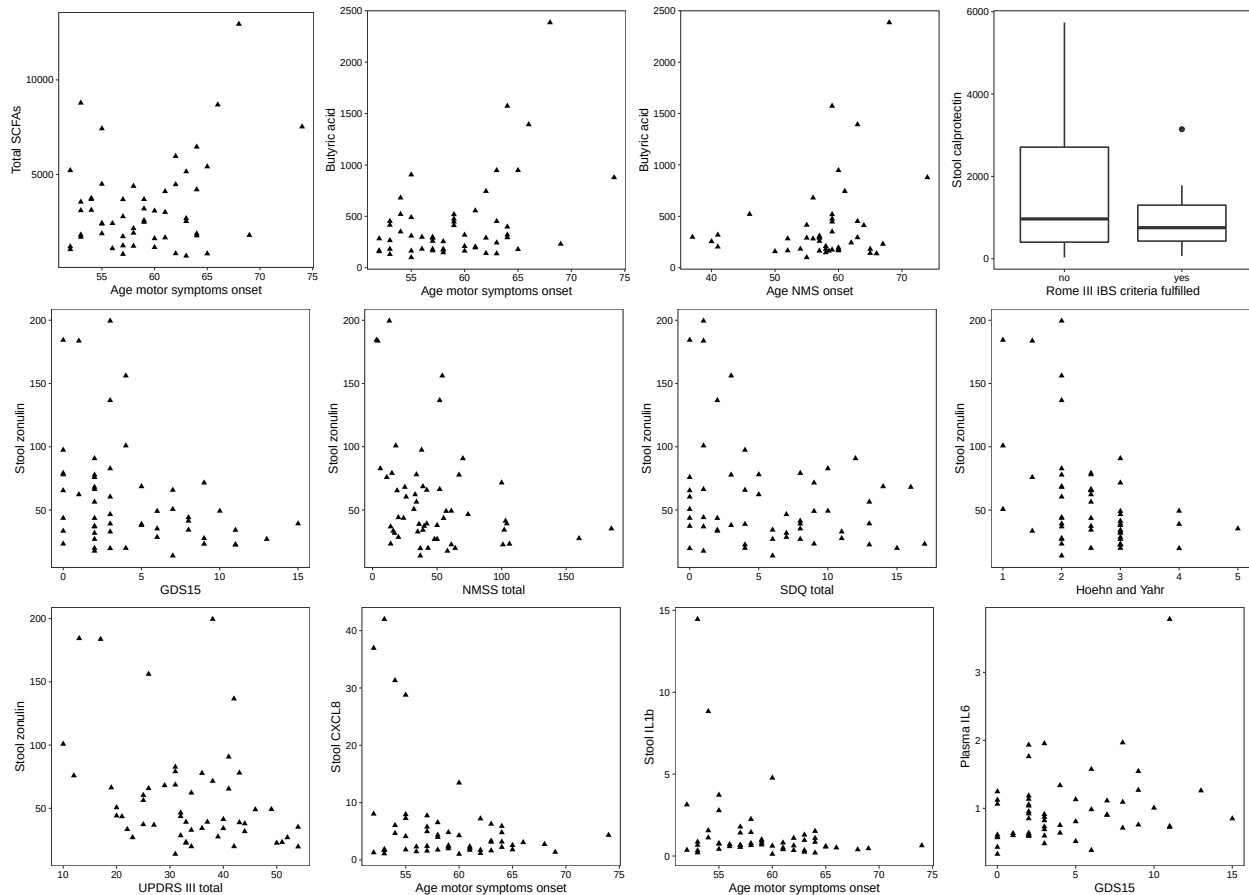
```
# Plot variables significant for PD-only
conf_cors_pd <- apply(subset(sig_cors, `PD patients` < 0.05 &
  is.na(`All subjects`)), 1,
  function(x){
    if(is.numeric(pdcy_meta[, x["Confounder_simple"]])) {
      basic_scatter_pd(pdcy_meta, x["Confounder_simple"],
```

```

x["variable"])] +
  xlab(x["Confounder"]) +
  ylab(inf_scfa_vars2[x["variable"]])
} else {
  basic_box_pd(
    pdcy_meta, x["Confounder_simple"],
    x["variable"]) +
    xlab(x["Confounder"]) +
    ylab(inf_scfa_vars2[x["variable"]])
  })
})

```

```
do.call("grid.arrange", c(conf_cors_pd, ncol = 4))
```



```
# Export both figures together
```

```
library("grid")
```

```
pdf("Outputs/figS2_clin_cors_scatters.pdf", width = 4.5, height = 7)
```

```

grid.arrange(
  textGrob("A", x = unit(0.025, "npc"), y = unit(0.4, "npc"),
    gp = gpar(fontface = "bold", fontsize = 12)),
  arrangeGrob(
    do.call("arrangeGrob",
      c(conf_cors_full[1:(length(conf_cors_full)-1)], ncol = 4)),
    arrangeGrob(clin_plot_legend, conf_cors_full[[length(conf_cors_full)]]),

```

```

      nrow = 1, widths = c(3, 1)),
      nrow = 2, heights = c(3, 1)),
      textGrob("B", x = unit(0.025, "npc"), y = unit(0.4, "npc"),
        gp = gpar(fontface = "bold", fontsize = 12)),
      do.call("arrangeGrob", c(conf_cors_pd, ncol = 4)),
      ncol = 1, heights = c(0.1, 1.35, 0.1, 1))
dev.off()

```

2.3.1.3 Linear regression corrected for age and/or time from disease onset

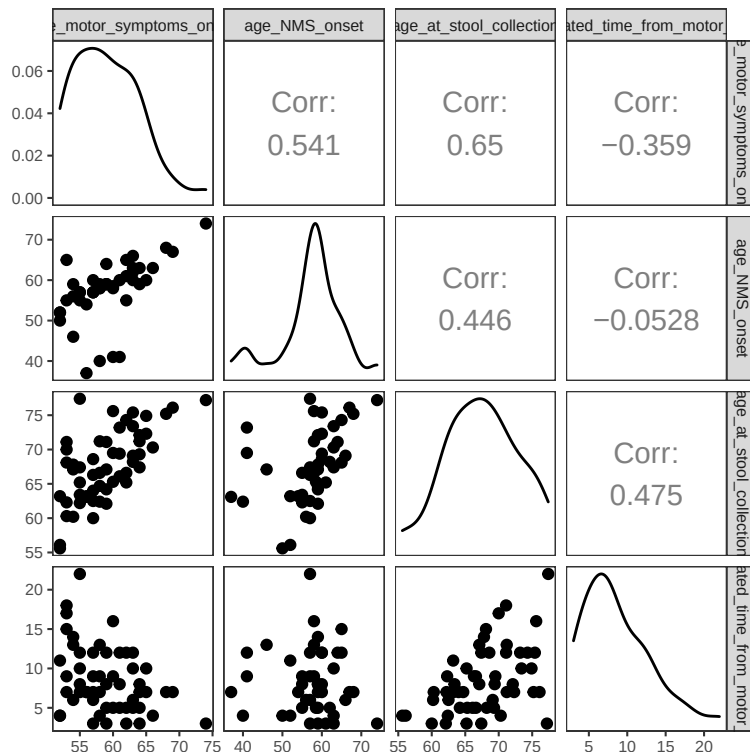
The PD-related correlations are interesting, but should be corrected for some confounding factors, mainly age and disease duration. Prior to corrections, check the potential confounding variables for collinearity.

Butyric acid ~ age motor symptoms onset:

```

ggpairs(subset(
  pdcy_meta, Group == "P"))[,c("age_motor_symptoms_onset",
    "age_NMS_onset",
    "age_at_stool_collection",
    "calculated_time_from_motor_onset")] +
  theme_bw(base_size = 8) +
  theme(panel.grid = element_blank())

```



All three variables are somewhat correlated; use the less collinear one (calculated time from motor onset) for correction.

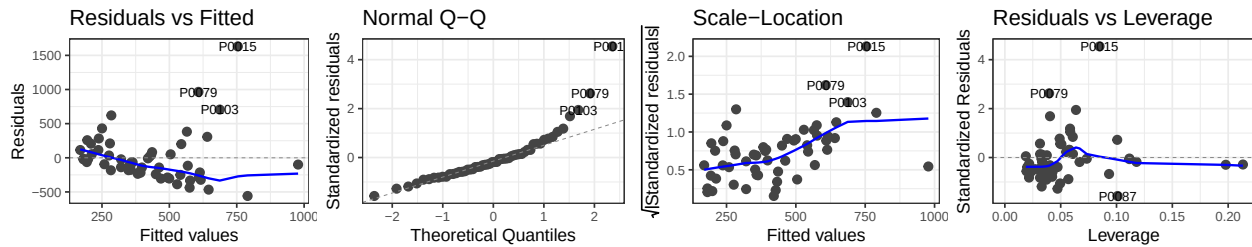
Model for butyric acid ~ time from motor onset: run and plot residuals.

```

buty_motors_regr <- lm(butyric_acid ~ age_motor_symptoms_onset +
  calculated_time_from_motor_onset,
  data = subset(pdcy_meta, Group == "P"))

```

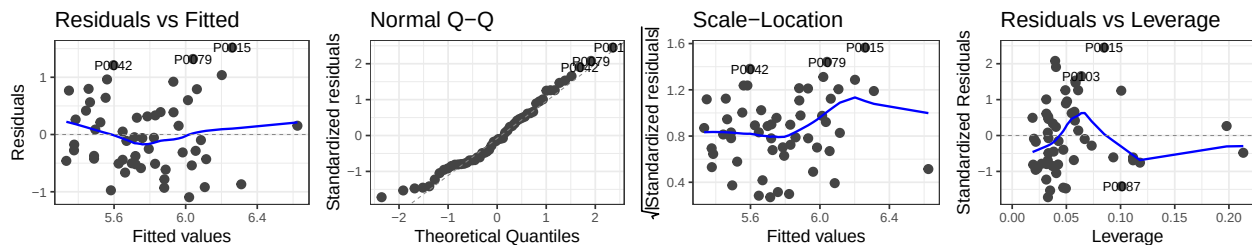
```
autoplot(buty_motors_regr, label.size = 2, nrow = 1) +
  theme_bw(base_size = 8)
```



The diagnostic plots don't look very good - log-transform butyric acid and plot again:

```
buty_motors_regr <- lm(log(butyric_acid) ~ age_motor_symptoms_onset +
  calculated_time_from_motor_onset,
  data = subset(pdcy_meta, Group == "P"))

autoplot(buty_motors_regr, label.size = 2, nrow = 1) +
  theme_bw(base_size = 8)
```



This looks better; use the log-transformed variable. Show model summary:

```
summary(buty_motors_regr)
```

```
##
## Call:
## lm(formula = log(butyric_acid) ~ age_motor_symptoms_onset + calculated_time_from_motor_onset,
##     data = subset(pdcy_meta, Group == "P"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.09495 -0.49156 -0.07992  0.39073  1.51757
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.89638    1.24023   2.335  0.0235 *
## age_motor_symptoms_onset  0.05098    0.01958   2.604  0.0121 *
## calculated_time_from_motor_onset -0.01471    0.02253  -0.653  0.5166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6478 on 51 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.1595, Adjusted R-squared:  0.1265
## F-statistic: 4.838 on 2 and 51 DF,  p-value: 0.01192
```

Age motor symptoms onset is still significant after correction.

What about age NMS onset (still using log-transformed butyric acid)?

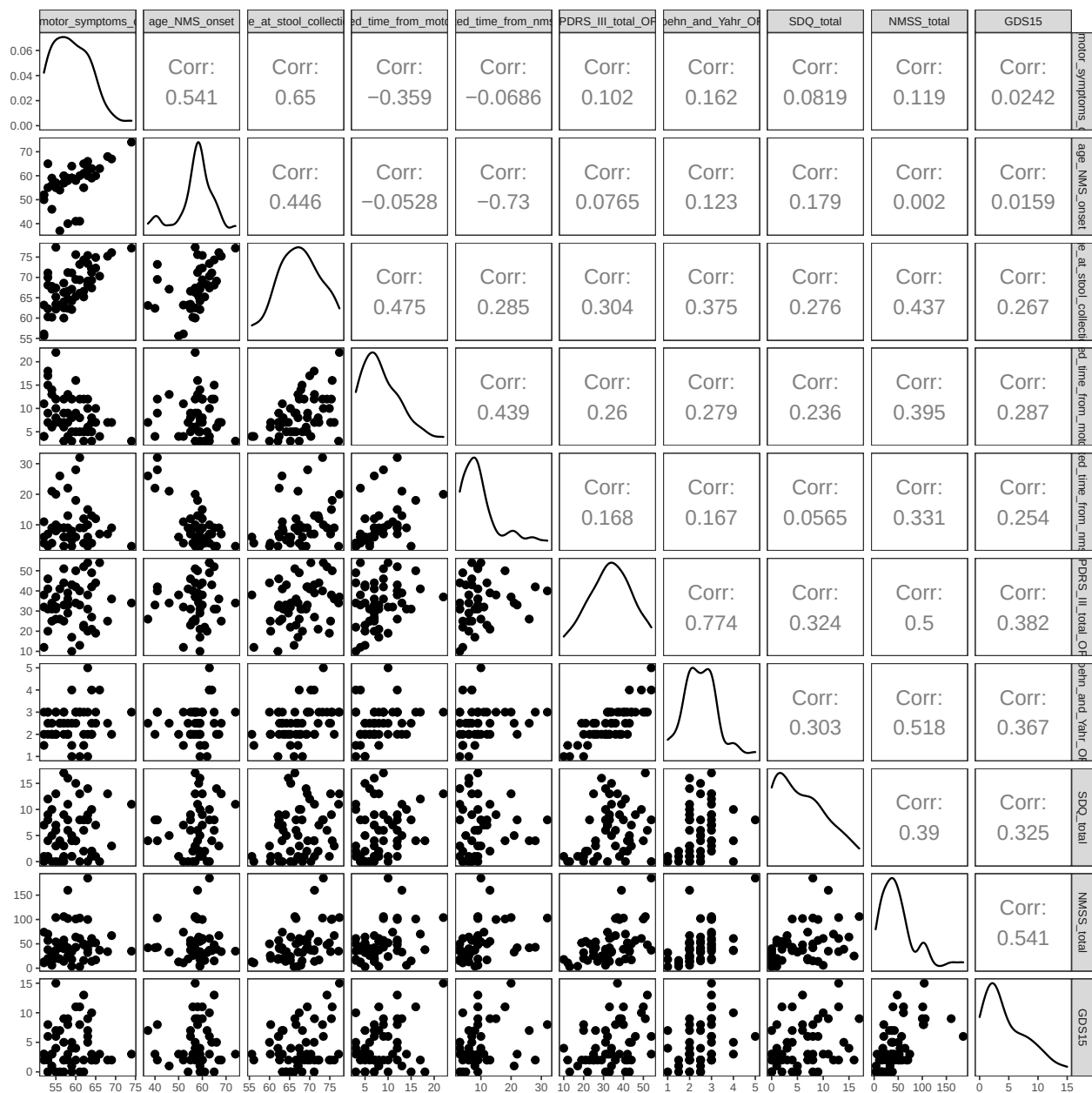
```
summary(lm(log(butyric_acid) ~ age_NMS_onset +
  calculated_time_from_motor_onset,
  data = subset(pdcy_meta, Group == "P")))
```

```
##
## Call:
## lm(formula = log(butyric_acid) ~ age_NMS_onset + calculated_time_from_motor_onset,
##     data = subset(pdcy_meta, Group == "P"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.0909 -0.5257 -0.1332  0.3247  1.7834
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.88412    0.84704   5.766 9.3e-07 ***
## age_NMS_onset      0.01996    0.01399   1.427  0.161
## calculated_time_from_motor_onset -0.03539    0.02517  -1.406  0.167
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.687 on 41 degrees of freedom
## (11 observations deleted due to missingness)
## Multiple R-squared:  0.09366,    Adjusted R-squared:  0.04945
## F-statistic: 2.119 on 2 and 41 DF,  p-value: 0.1332
```

Age NMS onset is not significant in this model.

Stool zonulin and PD-related variables + potential confounders: check for collinearity:

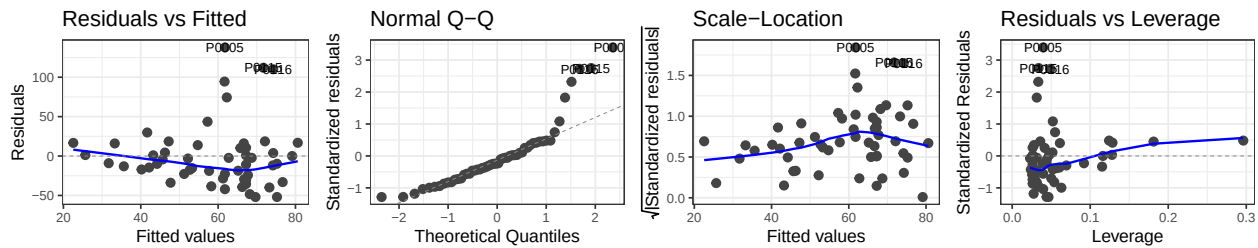
```
ggpairs(subset(
  pdcy_meta, Group == "P"), c("age_motor_symptoms_onset",
  "age_NMS_onset",
  "age_at_stool_collection",
  "calculated_time_from_motor_onset",
  "calculated_time_from_nms_onset",
  "UPDRS_III_total_OFF",
  "Hoehn_and_Yahr_OFF",
  "SDQ_total",
  "NMSS_total",
  "GDS15")) +
  theme_bw(base_size = 8) +
  theme(panel.grid = element_blank())
```



The variables of interest are slightly correlated with the age variables, but not very highly.

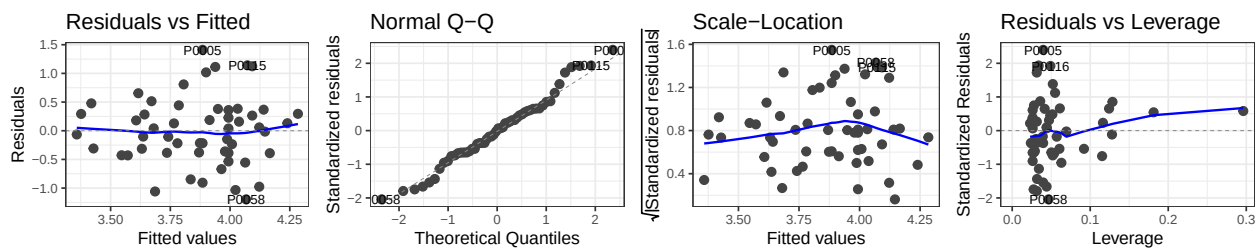
Check residual plots for one example model:

```
autoplot(lm(Szonulin ~ GDS15 +
  calculated_time_from_motor_onset,
  data = subset(pdcy_meta, Group == "P")),
  label.size = 2, nrow = 1) +
  theme_bw(base_size = 8)
```

Similar issues to above; try with log transformation of zonulin:

```
autoplot(lm(log(Szonulin) ~ GDS15 +
  calculated_time_from_motor_onset,
  data = subset(pdcy_meta, Group == "P")),
  label.size = 2, nrow = 1) +
  theme_bw(base_size = 8)
```



Though not perfect, this is better; use log-transformed zonulin in all models.

Run regression for all variables of interest with log-transformed zonulin, corrected for calculated time from motor onset:

```
szonulin_motors_regr <- as.data.frame(matrix(ncol = 4, nrow = 0))
colnames(szonulin_motors_regr) <- c("Model", "pValModel",
  "pValVar", "pValConf")

for(i in c("UPDRS_III_total_OFF", "Hoehn_and_Yahr_OFF",
  "SDQ_total", "NMSS_total", "GDS15")){
  m <- as.formula(paste("log(Szonulin) ~", i,
    "+calculated_time_from_motor_onset"))
  res <- summary(lm(m, subset(pdcy_meta, Group == "P")))
  szonulin_motors_regr <- rbind(szonulin_motors_regr,
    data.frame(Model = deparse(m),
      pValModel =
        unname(pf(res$fstatistic[1],
          res$fstatistic[2],
          res$fstatistic[3],
          lower.tail=FALSE)),
      pValVar1 = res$coefficients[2, 4],
      pValVar2 = res$coefficients[3, 4]))
}

# Show results
kable_styling(kable(szonulin_motors_regr, digits = 4,
  booktabs = TRUE, linesep = "",
  row.names = FALSE), font_size = 8)
```

Model	pValModel	pValVar1	pValVar2
log(Szonulin) ~ UPDRS_III_total_OFF + calculated_time_from_motor_onset	0.0166	0.0045	0.6006
log(Szonulin) ~ Hoehn_and_Yahr_OFF + calculated_time_from_motor_onset	0.0068	0.0017	0.5023
log(Szonulin) ~ SDQ_total + calculated_time_from_motor_onset	0.1517	0.0534	0.6846
log(Szonulin) ~ NMSS_total + calculated_time_from_motor_onset	0.0465	0.0137	0.3459
log(Szonulin) ~ GDS15 + calculated_time_from_motor_onset	0.0309	0.0087	0.4725

All five variables remain significant after correction (and calculated time from motor onset is not significant).

Same, but corrected for age from non-motor symptom onset instead:

```
szonulin_nms_regr <- as.data.frame(matrix(ncol = 4, nrow = 0))
colnames(szonulin_nms_regr) <- c("Model", "pValModel",
                                "pValVar", "pValConf")

for(i in c("UPDRS_III_total_OFF", "Hoehn_and_Yahr_OFF",
          "SDQ_total", "NMSS_total", "GDS15")){
  m <- as.formula(paste("log(Szonulin) ~", i,
                        "+calculated_time_from_nms_onset"))
  res <- summary(lm(m, subset(pdcy_meta, Group == "P")))
  szonulin_nms_regr <- rbind(szonulin_nms_regr,
                             data.frame(Model = deparse(m),
                                         pValModel =
                                           unname(pf(res$fstatistic[1],
                                                    res$fstatistic[2],
                                                    res$fstatistic[3],
                                                    lower.tail=FALSE))),
                             pValVar1 = res$coefficients[2, 4],
                             pValVar2 = res$coefficients[3, 4]))
}

# Show results
kable_styling(kable(szonulin_nms_regr, digits = 4,
                    booktabs = TRUE, linesep = "",
                    row.names = FALSE), font_size = 8)
```

Model	pValModel	pValVar1	pValVar2
log(Szonulin) ~ UPDRS_III_total_OFF + calculated_time_from_nms_onset	0.0133	0.0079	0.3955
log(Szonulin) ~ Hoehn_and_Yahr_OFF + calculated_time_from_nms_onset	0.0159	0.0096	0.3902
log(Szonulin) ~ SDQ_total + calculated_time_from_nms_onset	0.1762	0.1473	0.2710
log(Szonulin) ~ NMSS_total + calculated_time_from_nms_onset	0.0429	0.0270	0.6807
log(Szonulin) ~ GDS15 + calculated_time_from_nms_onset	0.0821	0.0575	0.4954

3 Comparisons with microbiota data

Import microbiota data and combine with the metadata:

```
# Import
pd_otus <- as.matrix(read.csv("Inputs/pdfu_16s_OTU_counts.csv", row.names = 1))
pd_tax <- as.matrix(read.csv("Inputs/pdfu_16s_taxonomy.csv", row.names = 1))

# Trim the OTU table to follow-up only
pd_otus <- pd_otus[grepl("N", rownames(pd_otus)), ]
```

```

# Remove timepoint identifier from row names
rownames(pd_otus) <- sub("N", "", rownames(pd_otus))

# Match to the samples that are in the metadata
pd_otus <- pd_otus[rownames(pdcy_meta), ]

# Combine data into a phyloseq object
pdcy_phy <- phyloseq(otu_table(t(pd_otus), taxa_are_rows = TRUE),
                     tax_table(pd_tax),
                     sample_data(pdcy_meta))

# Trim out OTUs that have no reads with this subset of samples
pdcy_phy <- subset_taxa(pdcy_phy, taxa_sums(pdcy_phy) > 0)

```

Make summarized objects on different taxonomic levels:

```

# Function for merging phyloseq data to different taxonomic levels:
collapseTaxLevel <- function(phylo_obj, level){

  collapseOtuTable <- function(phylo_obj, level){
    tax <- as.data.frame(as(tax_table(phylo_obj), "matrix"))
    otu <- as.data.frame(as(otu_table(phylo_obj), "matrix"))
    otu[, level] <- tax[, level]
    otu_collapsed <- melt(otu, id = (level))
    otu_collapsed <- acast(otu_collapsed,
                          as.formula(paste0(level, "~variable")), sum)
    otu_collapsed <- otu_collapsed[order(rownames(otu_collapsed)), ]
    return(otu_collapsed)
  }

  levelNum <- grep(level, colnames(tax_table(phylo_obj)))

  tax <- as(tax_table(phylo_obj), "matrix")
  tax <- apply(tax, 2, function(x) gsub(".*_unclassified", "unclassified", x))
  phylo_obj <- phyloseq(otu_table(phylo_obj),
                       sample_data(phylo_obj),
                       tax_table(tax))

  otu_collapsed <- collapseOtuTable(phylo_obj, level)

  tax <- as(tax_table(phylo_obj), "matrix")
  tax_collapsed <- unique(tax[, 1:levelNum])

  tax_collapsed <- tax_collapsed[-grep("unclassified", tax_collapsed[, levelNum]), ]
  tax_collapsed <- rbind(tax_collapsed, rep("unclassified", (levelNum)))

  rownames(tax_collapsed) <- tax_collapsed[, level]
  tax_collapsed <- tax_collapsed[order(rownames(tax_collapsed)), ]

  new_phylo_obj <- phyloseq(otu_table(otu_collapsed, taxa_are_rows = TRUE),
                           tax_table(tax_collapsed),
                           sample_data(phylo_obj))

  return(new_phylo_obj)
}

```

```

# Make genus and family level summarized phyloseq objects:
pdcy_phy_gen <- collapseTaxLevel(pdcy_phy, "Genus")
pdcy_phy_fam <- collapseTaxLevel(pdcy_phy, "Family")

# Subsets of data for downstream comparisons:

# PD-only
pdcy_phy_P <- subset_samples(pdcy_phy, Group == "P")
pdcy_phy_P_gen <- collapseTaxLevel(pdcy_phy_P, "Genus")
pdcy_phy_P_fam <- collapseTaxLevel(pdcy_phy_P, "Family")

# Control-only
pdcy_phy_C <- subset_samples(pdcy_phy, Group == "C")
pdcy_phy_C_gen <- collapseTaxLevel(pdcy_phy_C, "Genus")
pdcy_phy_C_fam <- collapseTaxLevel(pdcy_phy_C, "Family")

```

3.1 Alpha diversity

Look for associations between the markers/SCFAs and measures of alpha diversity (within-sample diversity) for two different indices: Shannon and inverse Simpson (both measuring richness and evenness).

3.1.1 Full data (PD + C together)

Calculate alpha diversity index values for full data (subsampled to an even number of reads per sample), and plot the values to check for outliers (dashed line: 3rd quartile + 1.5*IQR).

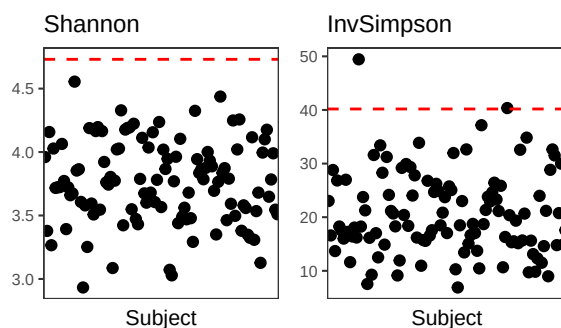
```

# Rarefy data for alpha diversity calculations
pdcy_phy_r <- rarefy_even_depth(pdcy_phy, rngseed = 2441662)

# Calculate alpha diversity values and add to metadata
pdcy_meta <- cbind(pdcy_meta,
  estimate_richness(pdcy_phy_r, measures = c("Shannon", "InvSimpson")))

# Check what the values look like (to screen for potential outliers etc)
adiv_check <- lapply(c("Shannon", "InvSimpson"), function(x) ol_plot(pdcy_meta, x))
do.call("grid.arrange", c(adiv_check, nrow = 1))

```



One subject had a clearly higher inverse Simpson value than everyone else; replace this subjects' alpha diversity estimates with NAs to exclude them from the comparisons.

```

pdcy_meta[pdcy_meta$InvSimpson == max(pdcy_meta$InvSimpson),
  c("Shannon", "InvSimpson")] <- NA

```

Calculate correlations for alpha diversity vs markers/SCFAs and show results significant for at least one

diversity measure:

```
# Function for calculating correlations
divCors <- function(df, varlist){
  res <- t(sapply(varlist, function(x) c(
    unname(unlist(cor.test(y = df[,x], x = df$Shannon)[c("estimate", "p.value")])),
    unname(unlist(cor.test(y = df[,x], x = df$InvSimpson)[c("estimate", "p.value")])))
  colnames(res) <- c("ShanCor", "ShanP", "InvSCor", "InvSP")
  res <- data.frame(Var = varlist, res)
  return(res)
}

# Calculate correlations
adiv_cors_full <- divCors(pdcy_meta, inf_scfa_vars)

# Significant results
kable_styling(kable(subset(adiv_cors_full, rowSums(adiv_cors_full[, c(3, 5)] < 0.05) > 0),
  booktabs = TRUE, linesep = "", digits = 4, row.names = FALSE), font_size = 8)
```

Var	ShanCor	ShanP	InvSCor	InvSP
total_SCFAs	-0.2091	0.0283	-0.1722	0.0720
propionic_acid	-0.2363	0.0129	-0.1694	0.0768
butyric_acid	-0.1950	0.0412	-0.1907	0.0459
SNGAL	-0.3643	0.0001	-0.2787	0.0040
Scalprotectin	-0.2780	0.0034	-0.2325	0.0150
Szonulin	-0.2935	0.0019	-0.2279	0.0167
SIL1b	-0.1890	0.0490	-0.1898	0.0481

3.1.2 Control only / PD only

Calculate similar correlations separately for control only and PD only subsets and show significant results:

```
# C-only correlations
adiv_cors_C <- data.frame(divCors(subset(pdcy_meta, Group == "C"), inf_scfa_vars),
  Set = "C_only")

# PD-only correlations
adiv_cors_PD <- data.frame(divCors(subset(pdcy_meta, Group == "P"), inf_scfa_vars),
  Set = "P_only")

# Combine
adiv_cors_sep <- rbind(adiv_cors_C, adiv_cors_PD)

# Order by variable
adiv_cors_sep <- adiv_cors_sep[order(adiv_cors_sep$Var), ]

# Variables with significant hits for either subset
adiv_cors_sep_sigs <- as.character(unique(
  adiv_cors_sep[which(rowSums(adiv_cors_sep[, c(3, 5)] < 0.05) > 0), "Var"]))

# Results for the variables that are significant for either subset
kable_styling(kable(subset(adiv_cors_sep, Var %in% adiv_cors_sep_sigs), booktabs = TRUE,
  linesep = "", digits = 4, row.names = FALSE), font_size = 8)
```

Var	ShanCor	ShanP	InvSCor	InvSP	Set
acetic_acid	-0.3717	0.0052	-0.3233	0.0160	C_only
acetic_acid	-0.0175	0.8993	0.0001	0.9995	P_only
butyric_acid	-0.3165	0.0186	-0.2873	0.0334	C_only
butyric_acid	-0.1097	0.4254	-0.1203	0.3818	P_only
propionic_acid	-0.4445	0.0007	-0.3755	0.0047	C_only
propionic_acid	-0.0119	0.9313	0.0456	0.7410	P_only
Scalprotectin	-0.2191	0.1115	-0.2476	0.0711	C_only
Scalprotectin	-0.3075	0.0224	-0.2298	0.0914	P_only
SCXCL8	0.0427	0.7569	-0.0867	0.5291	C_only
SCXCL8	-0.2825	0.0366	-0.2453	0.0711	P_only
SNGAL	-0.4994	0.0001	-0.4213	0.0015	C_only
SNGAL	-0.2047	0.1496	-0.1497	0.2943	P_only
Szonulin	-0.3648	0.0062	-0.1952	0.1533	C_only
Szonulin	-0.2412	0.0760	-0.2473	0.0687	P_only
total_SCFAs	-0.4105	0.0019	-0.3513	0.0085	C_only
total_SCFAs	-0.0285	0.8364	-0.0210	0.8788	P_only

3.1.2.1 Fig S3

Make a figure summarizing the alpha diversity correlation results:

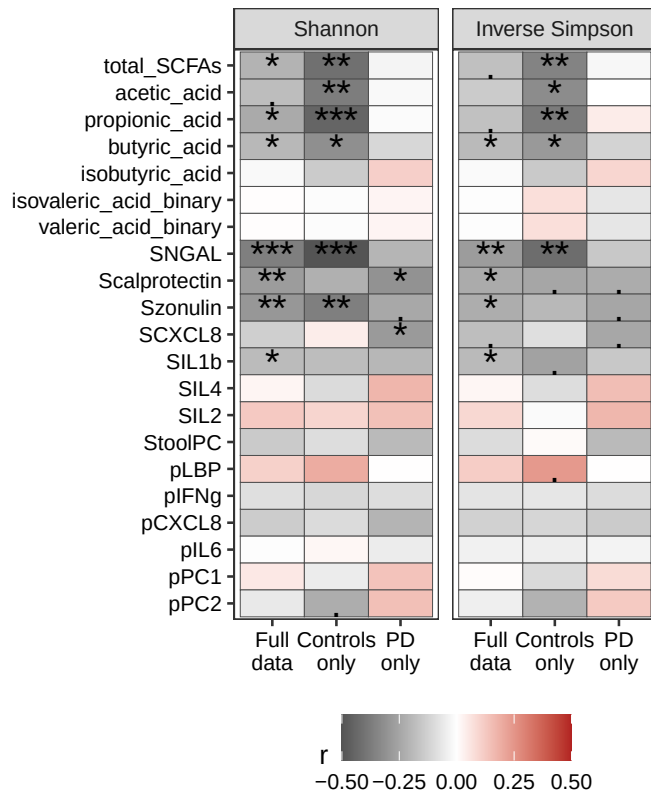
```
# Combine the results for full + subset
adiv_cors_full$Set <- "Full"
adiv_cors <- melt(rbind(adiv_cors_full, adiv_cors_sep))

# Split some variables
adiv_cors$Value <- sub(".*[nS]", "", adiv_cors$variable)
adiv_cors$Div <- sub("[PC].*", "", adiv_cors$variable)
adiv_cors$Set <- factor(adiv_cors$Set, levels = c("Full", "C_only", "P_only"))

# Reorganize data.frame
adiv_cors_df <- dcast(adiv_cors, Var + Set + Div ~ Value)

# Reorder variables for nicer plotting
adiv_cors_df$Var <- factor(adiv_cors_df$Var, levels = rev(inf_scfa_vars))
adiv_cors_df$Div <- factor(adiv_cors_df$Div, levels = c("Shan", "InvS"))
levels(adiv_cors_df$Div) <- c("Shannon", "Inverse Simpson")
levels(adiv_cors_df$Set) <- c("Full\ndata", "Controls\nonly", "PD\nonly")

adiv_cors_plot <- ggplot(adiv_cors_df, aes(x = Set, y = Var, fill = Cor)) +
  geom_tile(color = "gray30") +
  theme_bw(base_size = 10) +
  facet_grid(~Div) +
  scale_fill_gradient2(low = "gray33", mid = "white", high = "firebrick",
    name = "r", limits = c(-0.5, 0.5)) +
  geom_text(aes(label = stars.pval(P)), size = 5) +
  theme(legend.position = "bottom",
    axis.text = element_text(color = "black"),
    axis.title = element_blank(),
    panel.grid = element_blank())
adiv_cors_plot
```



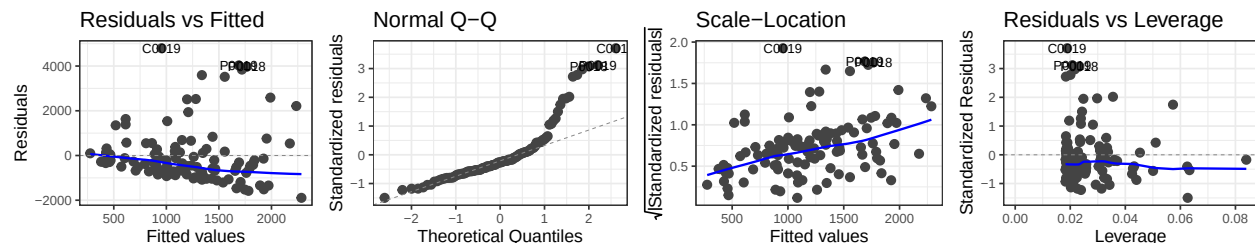
```
ggsave(adv_cors_plot, filename = "Outputs/figS3_adv_cors.pdf",
       width = maxwd*0.6, height = maxhi*0.6, units = "mm")
```

3.1.3 Corrected for PD vs C

To further explore the relationship of diversity, markers/SCFAs and PD, run linear regressions with the marker/SCFA as the dependent variable and diversity index + Group as explanatory variables. Test with a few example cases:

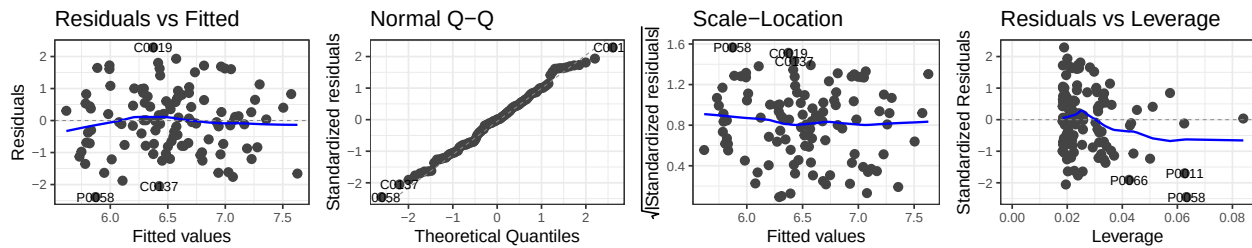
```
# Make model
lm_calpro_shan <- lm(Scalprotectin ~ Shannon + Group, pdcy_meta)

# Residual plots
autoplot(lm_calpro_shan, label.size = 2, nrow = 1) + theme_bw(base_size = 8)
```



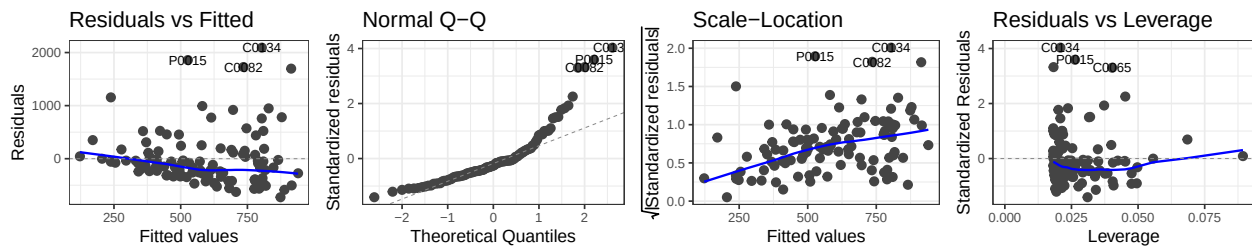
The diagnostic plots for the residuals imply heteroscedasticity. Log-transform calprotectin and plot again:

```
lm_shan_calpro_2 <- lm(log(Scalprotectin) ~ Shannon + Group, pdcy_meta)
autoplot(lm_shan_calpro_2, label.size = 2, nrow = 1) + theme_bw(base_size = 8)
```



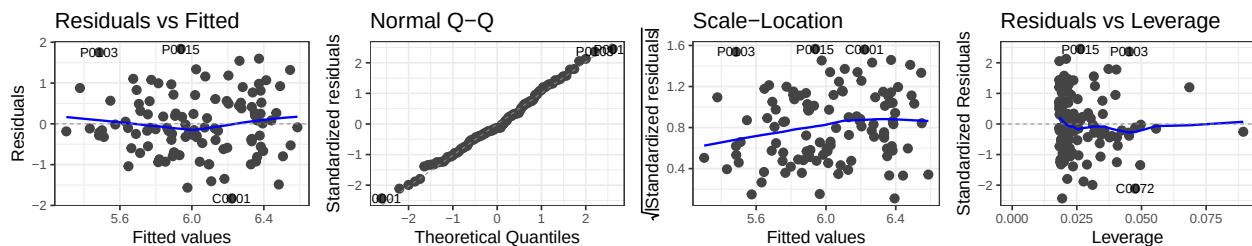
Run a similar test for Butyric acid ~ inverse Simpson + Group:

```
lm_buty_shan <- lm(butyric_acid ~ InvSimpson + Group, pdcy_meta)
autoplot(lm_buty_shan, label.size = 2, nrow = 1) + theme_bw(base_size = 8)
```



Try with log-transformed SCFA data:

```
lm_buty_shan2 <- lm(log(butyric_acid) ~ InvSimpson + Group, pdcy_meta)
autoplot(lm_buty_shan2, label.size = 2, nrow = 1) + theme_bw(base_size = 8)
```



Since the log-transformed response variables seem to provide better results, run all regression testing with log-transformed response variables. Test both diversity measures for the markers that were significant for full data for at least one measure in the previous regressions. Include additional correction for sex, since it seemed to have a strong effect on the PD-differences for some SCFAs and markers.

```
lm_vars <- as.character(unique(subset(adv_cors,
                                     value < 0.05 & Value == "P" & Set == "Full")$Var))

pdc_div_lms <- as.data.frame(matrix(ncol = 4, nrow = 0))
colnames(pdc_div_lms) <- c("Model", "pValModel", "pValGroup", "pValDiv")

for(i in lm_vars){
  for(j in c("Shannon", "InvSimpson")){
    m <- as.formula(paste("log(", i, ")~sex+Group+", j))
    res <- summary(lm(m, pdcy_meta))
    pdc_div_lms <- rbind(pdc_div_lms,
                        data.frame(Model = deparse(m),
                                   pValModel = unname(pf(res$fstatistic[1],
                                                           res$fstatistic[2],
                                                           res$fstatistic[3],
                                                           lower.tail=FALSE)),
```



```

    pValSex = res$coefficients[2, 4],
    pValGroup = res$coefficients[3, 4],
    pValDiv = res$coefficients[4, 4]))
  }
}

# Show results
kable_styling(kable(pdc_div_lms, digits = 4, booktabs = TRUE, linesep = "",
                    row.names = FALSE), font_size = 8)

```

Model	pValModel	pValSex	pValGroup	pValDiv
log(total_SCFAs) ~ sex + Group + Shannon	0.0101	0.3113	0.0250	0.0089
log(total_SCFAs) ~ sex + Group + InvSimpson	0.0362	0.3486	0.0414	0.0434
log(propionic_acid) ~ sex + Group + Shannon	0.0057	0.5977	0.0196	0.0039
log(propionic_acid) ~ sex + Group + InvSimpson	0.0452	0.6625	0.0365	0.0501
log(butyric_acid) ~ sex + Group + Shannon	0.0002	0.2563	0.0002	0.0058
log(butyric_acid) ~ sex + Group + InvSimpson	0.0006	0.2855	0.0005	0.0184
log(SNGAL) ~ sex + Group + Shannon	0.0000	0.9947	0.3654	0.0000
log(SNGAL) ~ sex + Group + InvSimpson	0.0017	0.8792	0.2559	0.0002
log(Scalprotectin) ~ sex + Group + Shannon	0.0000	0.0208	0.0620	0.0000
log(Scalprotectin) ~ sex + Group + InvSimpson	0.0000	0.0232	0.0327	0.0000
log(Szonulin) ~ sex + Group + Shannon	0.0104	0.8834	0.8230	0.0010
log(Szonulin) ~ sex + Group + InvSimpson	0.0462	0.8295	0.6500	0.0059
log(SIL1b) ~ sex + Group + Shannon	0.4071	0.1299	0.4814	0.5642
log(SIL1b) ~ sex + Group + InvSimpson	0.2584	0.1251	0.5000	0.2287

3.1.3.1 Table S2A

```

# Fix variable labels and reorganize
pdc_div_lms$Model <- sub("\\(S", "\\(Stool ", sub("_acid", " acid", pdc_div_lms$Model))
pdc_div_lms$Variable <- capitalize(sub("\\).*", "", sub("log\\(", "", pdc_div_lms$Model)))
pdc_div_lms$DivIndex <- sub(".*\\+ ", "", pdc_div_lms$Model)
pdc_div_lms <- pdc_div_lms[, c("Model", "Variable", "DivIndex", "pValModel",
                              "pValSex", "pValGroup", "pValDiv")]

# Export
write.csv(pdc_div_lms, "Outputs/tableS2A.csv", row.names = FALSE)

```

Additionally, test a set of models with an interaction: log(marker or SCFA) ~ sex + Group * Diversity.

```

pdc_div_lms2 <- as.data.frame(matrix(ncol = 5, nrow = 0))
colnames(pdc_div_lms2) <- c("Model", "pValModel",
                           "pValGroup", "pValDiv", "pValGroupDiv")

for(i in lm_vars){
  for(j in c("Shannon", "InvSimpson")){
    m <- as.formula(paste("log(", i, ")~sex + Group *", j))
    res <- summary(lm(m, pdcy_meta))
    pdc_div_lms2 <- rbind(pdc_div_lms2,
                          data.frame(Model = deparse(m),
                                     pValModel = unname(pf(res$fstatistic[1],
                                                           res$fstatistic[2],
                                                           res$fstatistic[3],
                                                           lower.tail = FALSE)),
                                     pValSex = res$coefficients[2, 4],
                                     pValGroup = res$coefficients[3, 4],

```

```

    pValDiv = res$coefficients[4, 4],
    pValGroupDiv = res$coefficients[5, 4]))
  }
}

# Trim to those significant models where group and interaction are p < 0.1
pdc_div_sig_lms2 <- subset(pdc_div_lms2, pValModel < 0.05 &
  pValGroup < 0.1 & pValGroupDiv < 0.1)

# Show results
kable_styling(kable(pdc_div_sig_lms2, digits = 4, booktabs = TRUE,
  linesep = "", row.names = FALSE), font_size = 8)

```

Model	pValModel	pValSex	pValGroup	pValDiv	pValGroupDiv
log(total_SCFAs) ~ sex + Group * Shannon	0.0041	0.3462	0.0291	0.0012	0.0454
log(total_SCFAs) ~ sex + Group * InvSimpson	0.0161	0.3906	0.0137	0.0060	0.0567
log(propionic_acid) ~ sex + Group * Shannon	0.0012	0.6603	0.0117	0.0002	0.0197
log(propionic_acid) ~ sex + Group * InvSimpson	0.0079	0.7396	0.0033	0.0020	0.0166
log(butyric_acid) ~ sex + Group * Shannon	0.0002	0.2829	0.0473	0.0018	0.0941
log(SNGAL) ~ sex + Group * Shannon	0.0000	0.9085	0.0925	0.0000	0.0781

3.1.3.2 Table S2B

```

# Fix variable labels
pdc_div_lms2$Model <- sub("\\(S", "\\(Stool ", sub("_acid", " acid", pdc_div_lms2$Model))
pdc_div_lms2$Variable <- capitalize(
  sub("\\).*", "", sub("log\\(", "", pdc_div_lms2$Model)))
pdc_div_lms2$DivIndex <- sub(".*\\*", "", pdc_div_lms2$Model)
pdc_div_lms2 <- pdc_div_lms2[, c("Model", "Variable", "DivIndex", "pValModel",
  "pValSex", "pValGroup", "pValDiv", "pValGroupDiv")]

# Export
write.csv(pdc_div_lms2, "Outputs/tableS2B.csv", row.names = FALSE)

```

3.1.4 Analyses with additional confounders

Finally, explore a few of the variable of interest ~ PD/control + diversity models in more detail:

3.1.4.1 Calprotectin

Previous results for calprotectin models:

```

pdc_div_lms$pValGroupDiv <- NA
pdc_div_lm_all <- rbind(pdc_div_lms, pdc_div_lms2)

kable_styling(kable(pdc_div_lm_all[grepl("calpro", pdc_div_lm_all$Model), c(1, 4:8)],
  digits = 4, booktabs = TRUE, linesep = "", row.names = FALSE), font_size = 8)

```

Model	pValModel	pValSex	pValGroup	pValDiv	pValGroupDiv
log(Stool calprotectin) ~ sex + Group + Shannon	0	0.0208	0.0620	0.0000	NA
log(Stool calprotectin) ~ sex + Group + InvSimpson	0	0.0232	0.0327	0.0000	NA
log(Stool calprotectin) ~ sex + Group * Shannon	0	0.0196	0.3708	0.0133	0.4600
log(Stool calprotectin) ~ sex + Group * InvSimpson	0	0.0240	0.4905	0.0068	0.9875

Results were similar for the two indices; focus on inverse Simpson, the index with the lower p -value. The interaction was not significant for PD * diversity. As alternatives, since there was a notable sex difference for calprotectin, test additional model with interaction:

```
print(anova(lm(log(Scalprotectin) ~ Group + sex + InvSimpson, data = pdcy_meta),
            lm(log(Scalprotectin) ~ Group + sex * InvSimpson, data = pdcy_meta)))
```

```
## Analysis of Variance Table
##
## Model 1: log(Scalprotectin) ~ Group + sex + InvSimpson
## Model 2: log(Scalprotectin) ~ Group + sex * InvSimpson
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1     105 102.249
## 2     104  90.482   1    11.767 13.525 0.0003747 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In fact, it seems there could be an interaction for sex * diversity; continue with that model. Potential additional confounders with $p < 0.05$ for calprotectin based on the clinical variable correlations (Fig 2C):

```
calp_conf <- subset(meta_cors_st_P, variable == "Scalprotectin" &
                    value == TRUE & Set == "All subjects")$Confounder
calp_conf
```

```
## [1] History diabetes PD relative      RBDSQ
## 64 Levels: Age at stool collection BMI GDS15 ... UPDRS IV total
```

Test a model with all the potential confounders plus sex, then repeat with only significant confounders plus Group and inverse Simpson, and collect results into a table:

```
# Original model without any confounders
calp_m0 <- lm(log(Scalprotectin) ~ Group + sex * InvSimpson, pdcy_meta)

# All confounders
calp_m1 <- lm(log(Scalprotectin) ~ Group + sex * InvSimpson +
              history_diabetes + PD_relative + RBDSQ, pdcy_meta)

# Check results
summary(calp_m1)
```

```
##
## Call:
## lm(formula = log(Scalprotectin) ~ Group + sex * InvSimpson +
##     history_diabetes + PD_relative + RBDSQ, data = pdcy_meta)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6526 -0.6577 -0.1046  0.6943  2.0498
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.69449    0.36900   20.852 < 2e-16 ***
## GroupP          0.10771    0.21569    0.499 0.618612
## sexM          -1.26093    0.53096   -2.375 0.019446 *
## InvSimpson     -0.08800    0.01575   -5.589 1.95e-07 ***
## history_diabetes1 0.59511    0.28158    2.113 0.037022 *
## PD_relative1    0.46081    0.24181    1.906 0.059538 .
## RBDSQ          0.05664    0.03728    1.519 0.131784
```

```
## sexM:InvSimpson    0.08446    0.02456    3.439 0.000851 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8993 on 101 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.3741, Adjusted R-squared:  0.3307
## F-statistic: 8.623 on 7 and 101 DF,  p-value: 2.924e-08

# History of diabetes is significant and PD relative borderline, keep them:
calp_m2 <- lm(log(Scalprotectin) ~ Group + sex * InvSimpson +
              history_diabetes + PD_relative, pdcy_meta)

# Package for convenient table output
library("huxtable")

# Customized table function
custom_huxreg <- function(m0, m1, m2, cn){
  ht <- huxreg("Simple model" = m0,
               "All confounders" = m1,
               "Main confounders" = m2,
               coefs = cn, error_pos = "same",
               stars = c(`.` = 0.1, `*` = 0.05, `**` = 0.01, `***` = 0.001))
  position(ht) <- "left"
  ht <- theme_plain(ht)
  return(ht)
}

calp_coefs <- c('Sex' = "sexM",
                'PD vs control' = 'GroupP',
                'Inverse Simpson index' = 'InvSimpson',
                'Diabetes' = 'history_diabetes1',
                'Relative with PD' = 'PD_relative1',
                'RBDSQ score' = 'RBDSQ',
                'sex and inverse Simpson interaction ' = 'sexM:InvSimpson')

# Show results of each model side by side
custom_huxreg(calp_m0, calp_m1, calp_m2, calp_coefs)
```

	Simple model	All confounders	Main confounders
Sex	-1.461 ** (0.546)	-1.261 * (0.531)	-1.336 * (0.532)
PD vs control	0.381 * (0.179)	0.108 (0.216)	0.299 · (0.176)
Inverse Simpson index	-0.095 *** (0.016)	-0.088 *** (0.016)	-0.090 *** (0.016)
Diabetes		0.595 * (0.282)	0.622 * (0.283)
Relative with PD		0.461 · (0.242)	0.458 · (0.243)
RBDSQ score		0.057 (0.037)	
sex and inverse Simpson interaction	0.093 *** (0.025)	0.084 *** (0.025)	0.087 *** (0.025)
N	109	109	109
R2	0.307	0.374	0.360
logLik	-144.517	-138.938	-140.169
AIC	301.033	295.876	296.339

*** p < 0.001; ** p < 0.01; * p < 0.05; · p < 0.1.

```
# Export
quick_xlsx(custom_huxreg(calp_m0, calp_m1, calp_m2, calp_coefs),
            file = "Outputs/tableS2C.xlsx", open = FALSE)
```

3.1.4.2 Propionic acid

Previous results for propionic acid models:

```
kable_styling(kable(pdc_div_lm_all[grep("propionic", pdc_div_lm_all$Model), c(1, 4:8)],
                    digits = 4, booktabs = TRUE, linesep = "", row.names = FALSE), font_size = 8)
```

Model	pValModel	pValSex	pValGroup	pValDiv	pValGroupDiv
log(propionic acid) ~ sex + Group + Shannon	0.0057	0.5977	0.0196	0.0039	NA
log(propionic acid) ~ sex + Group + InvSimpson	0.0452	0.6625	0.0365	0.0501	NA
log(propionic acid) ~ sex + Group * Shannon	0.0012	0.6603	0.0117	0.0002	0.0197
log(propionic acid) ~ sex + Group * InvSimpson	0.0079	0.7396	0.0033	0.0020	0.0166

Focus on Shannon, which had the lower p -values. Which is better, model with an interaction or without?

```
print(anova(
  lm(log(propionic_acid) ~ sex + Group + Shannon, data = pdcy_meta),
  lm(log(propionic_acid) ~ sex + Group * Shannon, data = pdcy_meta)))
```

```
## Analysis of Variance Table
##
## Model 1: log(propionic_acid) ~ sex + Group + Shannon
## Model 2: log(propionic_acid) ~ sex + Group * Shannon
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1     106 35.901
## 2     105 34.080   1    1.8212 5.6112 0.01967 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with an interaction was better, so continue with that, and add correction for confounding variables. Potential confounders with $p < 0.05$ for propionic acid from the clinical variable correlations (Fig 2A):

```
prop_conf <- subset(meta_cors_sc_P, variable == "propionic_acid" &
                    value == TRUE & Set == "All subjects")$Confounder
prop_conf
```

```
## [1] GDS15                      Medication ACE AT1      Medication statin
## [4] NMSS total                  Rome III 9-15 sum score
## 64 Levels: Age at stool collection BMI GDS15 ... UPDRS IV total
```

Test a model with all potential confounders plus sex, then repeat with significant confounders plus Group and inverse Simpson:

```
# Original model
prop_m0 <- lm(log(propionic_acid) ~ sex + Group * Shannon, pdcy_meta)

# All confounders
prop_m1 <- lm(log(propionic_acid) ~ sex + Group * Shannon + GDS15 + meds_ACE_AT1 +
              meds_statin + NMSS_total + Rome_III_constip_defec_sumscore_9.15,
              pdcy_meta)

# Check results
summary(prop_m1)
```

```
##
## Call:
## lm(formula = log(propionic_acid) ~ sex + Group * Shannon + GDS15 +
##      meds_ACE_AT1 + meds_statin + NMSS_total + Rome_III_constip_defec_sumscore_9.15,
##      data = pdcy_meta)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -1.39621 -0.33566  0.03743  0.30615  1.34008
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   9.727701   0.911298  10.675 < 2e-16 ***
## sexM                        -0.025213   0.111227  -0.227  0.821136
## GroupP                      -2.792932   1.299102  -2.150  0.033975 *
## Shannon                     -0.854215   0.238488  -3.582  0.000529 ***
## GDS15                       -0.029178   0.022658  -1.288  0.200804
## meds_ACE_AT11                0.165356   0.116188   1.423  0.157798
## meds_statin1                 0.163215   0.127530   1.280  0.203573
## NMSS_total                   0.002473   0.002634   0.939  0.350221
## Rome_III_constip_defec_sumscore_9.15 -0.030991  0.013985  -2.216  0.028960 *
## GroupP:Shannon               0.736296   0.347652   2.118  0.036662 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5551 on 100 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.2372, Adjusted R-squared:  0.1685
## F-statistic: 3.454 on 9 and 100 DF, p-value: 0.0009582

# Significant confounders
prop_m2 <- lm(log(propionic_acid) ~ Group * Shannon +
              Rome_III_constip_defec_sumscore_9.15, pdcy_meta)

prop_coefs <- c('Sex' = "sexM",
               'PD vs control' = 'GroupP',
               'Shannon index' = 'Shannon',
               'GDS15 score' = 'GDS15',
               'ACE-AT1 medication' = 'meds_ACE_AT11',
               'Statin medication' = 'meds_statin1',
               'NMSS total score' = 'NMSS_total',
               'Rome III 9-15 sum score' =
                 'Rome_III_constip_defec_sumscore_9.15',
               'PD/C and Shannon interaction' = 'GroupP:Shannon')

# Show results
custom_huxreg(prop_m0, prop_m1, prop_m2, prop_coefs)

# Export
quick_xlsx(custom_huxreg(prop_m0, prop_m1, prop_m2, prop_coefs),
            file = "Outputs/tableS2D.xlsx", open = FALSE)
```

3.1.4.3 Fig 3

Plots for the SCFA/markers ~ diversity linear regressions:

	Simple model	All confounders	Main confounders
Sex	0.048 (0.109)	-0.025 (0.111)	
PD vs control	-3.313 * (1.292)	-2.793 * (1.299)	-3.332 ** (1.266)
Shannon index	-0.925 *** (0.243)	-0.854 *** (0.238)	-0.898 *** (0.239)
GDS15 score		-0.029 (0.023)	
ACE-AT1 medication		0.165 (0.116)	
Statin medication		0.163 (0.128)	
NMSS total score		0.002 (0.003)	
Rome III 9-15 sum score		-0.031 * (0.014)	-0.026 * (0.013)
PD/C and Shannon interaction	0.812 * (0.343)	0.736 * (0.348)	0.853 * (0.336)
N	110	110	110
R2	0.156	0.237	0.188
logLik	-91.635	-86.098	-89.501
AIC	195.271	194.196	191.003

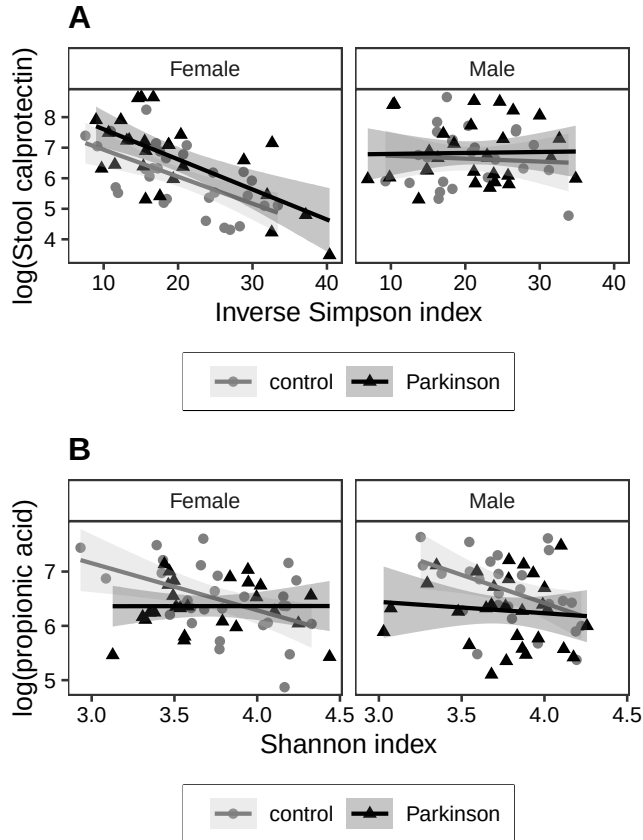
*** p < 0.001; ** p < 0.01; * p < 0.05; · p < 0.1.

```

# Custom scatterplot function
custom_lm_pd_s <- function(df, xvar, yvar){
  ggplot(df, aes_string(x = xvar, y = yvar,
                        color = "Group", shape = "Group", fill = "Group")) +
    geom_point(size = 1.5) +
    facet_grid(~sex,
              labeller = labeller(sex = setNames(c("Female", "Male"), c("F", "M")))) +
    scale_color_manual(values = c("gray50", "black"), label = pdc_labels) +
    scale_shape_manual(values = c(16, 17), label = pdc_labels) +
    scale_fill_manual(values = c("gray75", "gray25"), label = pdc_labels) +
    theme_bw(base_size = 10) +
    geom_smooth(method = "lm", size = 0.75, alpha = 0.3) +
    theme(panel.grid = element_blank(),
          axis.text = element_text(color = "black"),
          legend.position = "bottom",
          legend.key.height = unit(4, "mm"),
          legend.title = element_blank(),
          plot.title = element_text(face = "bold"),
          strip.background = element_rect(fill = "white"),
          legend.box.background = element_rect(color = "black"))
}

# Put together figure
pdc_lm_plots <- arrangeGrob(
  custom_lm_pd_s(pdcy_meta, "InvSimpson", "log(Scalprotectin)") +
    ylab("log(Stool calprotectin)") +
    xlab("Inverse Simpson index") +
    ggtitle("A"),
  custom_lm_pd_s(pdcy_meta, "Shannon", "log(propionic_acid)") +
    ylab("log(propionic acid)") +
    xlab("Shannon index") +
    ggtitle("B"),
  ncol = 1)
grid.arrange(pdc_lm_plots)

```



```
# Export
ggsave(pdc_lm_plots, filename = "Outputs/fig3_adiv_lms.tiff",
       width = 90, height = 125, units = "mm", dpi = 600,
       type = "cairo", compression = "lzw")
```

3.2 Beta diversity

Test beta diversity differences based on Bray-Curtis dissimilarities of subsampled data with adonis (PER-MANOVA). This test does not accept missing values, so drop subjects with NAs from the comparisons for each variable.

3.2.1 Full data (PD + C together)

```
# Run adonis for each variable separately
bdiv_all <- list()

for(i in inf_scfa_vars){
  phy <- subset_samples(pdcy_phy, !is.na(sample_data(pdcy_phy)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 1942890)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_all[[i]] <- adonis(as.formula(paste("dis ~", i)),
                        data = as(sample_data(rar), "data.frame"),
                        permutations = 9999)$aov.tab
}

# Grab p-values
bdiv_all_p <- sapply(inf_scfa_vars, function(x) bdiv_all[[x]][["Pr(>F)"]][1])
```



```
# Make data frame
bdiv_all_p <- data.frame(Variable = names(bdiv_all_p),
                        pVar = bdiv_all_p)
```

Significant results for these comparisons:

```
kable(subset(bdiv_all_p, pVar < 0.05), row.names = FALSE, booktabs = TRUE,
      linesep = "", digits = 3)
```

Variable	pVar
total_SCFAs	0.000
acetic_acid	0.000
propionic_acid	0.000
butyric_acid	0.000
SNGAL	0.008
Szonulin	0.001
SIL2	0.030
StoolPC	0.016

3.2.2 Corrected for PD vs control

Run adonis2 models with the immune marker/SCFA variables corrected for Group, since based on previous analyses, there is a significant beta diversity difference between PD and control.

```
# Run adonis for each variable
bdiv_pdc <- list()

for(i in inf_scfa_vars){
  phy <- subset_samples(pdcy_phy, !is.na(sample_data(pdcy_phy)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 1942890)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_pdc[[i]] <- adonis2(as.formula(paste("dis ~ Group + ", i)),
                        data = as(sample_data(rar), "data.frame"),
                        permutations = 9999, by = "margin")["Pr(>F)"]
}

# Grab p-values into data frame
bdiv_pdc_p <- data.frame(
  Var = inf_scfa_vars,
  pGroup = sapply(inf_scfa_vars, function(x) bdiv_pdc[[x]][1,]),
  pVar = sapply(inf_scfa_vars, function(x) bdiv_pdc[[x]][2,]))
```

Variables that were significant when corrected for Group:

```
kable(subset(bdiv_pdc_p, pVar < 0.05), row.names = FALSE, booktabs = TRUE,
      linesep = "", digits = 3)
```

Var	pGroup	pVar
total_SCFAs	0.001	0.000
acetic_acid	0.000	0.000
propionic_acid	0.001	0.000
butyric_acid	0.002	0.000
SNGAL	0.001	0.007
Szonulin	0.000	0.000
SIL2	0.000	0.030
StoolPC	0.000	0.022

3.2.3 Control only / PD only

As an alternative PD/control comparison, run tests separately for PD-only and control-only subsets.

3.2.3.0.1 C-only

```
bdiv_c <- list()

for(i in inf_scfa_vars){
  phy <- subset_samples(pdcy_phy_C, !is.na(sample_data(pdcy_phy_C)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 3656154)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_c[[i]] <- adonis(as.formula(paste("dis ~", i)),
                      data = as(sample_data(rar), "data.frame"),
                      permutations = 9999)$aov.tab
}

# Grab p-values
bdiv_c_p <- sapply(inf_scfa_vars, function(x) bdiv_c[[x]][["Pr(>F)"]][1])

# Make data frame
bdiv_c_p <- data.frame(Var = names(bdiv_c_p),
                      pVar = bdiv_c_p)
```

3.2.3.0.2 PD-only

```
bdiv_pd <- list()

for(i in inf_scfa_vars){
  phy <- subset_samples(pdcy_phy_P, !is.na(sample_data(pdcy_phy_P)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 2160957)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_pd[[i]] <- adonis(as.formula(paste("dis ~", i)),
                      data = as(sample_data(rar), "data.frame"),
                      permutations = 9999)$aov.tab
}

# Grab p-values
bdiv_pd_p <- sapply(inf_scfa_vars, function(x) bdiv_pd[[x]][["Pr(>F)"]][1])

# Make data frame
bdiv_pd_p <- data.frame(Var = names(bdiv_pd_p),
                      pVar = bdiv_pd_p)
```

Results with a significant uncorrected p -value in either the C-only or PD-only comparisons:

```
bdiv_pdc_separated <- data.frame(Var = rownames(bdiv_c_p),
                                pVar_C = bdiv_c_p[,2],
                                pVar_P = bdiv_pd_p[, 2])

kable(bdiv_pdc_separated[rowSums(bdiv_pdc_separated[, 2:3] < 0.05) > 0,],
      booktabs = TRUE, linesep = "", digits = 3, row.names = FALSE)
```

Var	pVar_C	pVar_P
total_SCFAs	0.000	0.066
acetic_acid	0.000	0.275
propionic_acid	0.001	0.217
butyric_acid	0.000	0.004
SNGAL	0.004	0.307
Szonulin	0.008	0.021

3.2.4 Analyses with additional confounders

Run the tests with additional correction for sex and the main beta diversity confounders from the follow-up microbiome article (Aho et al. 2019).

```
# Run adonis for each variable
bdiv_pdc_conf <- list()

for(i in inf_scfa_vars){
  phy <- subset_samples(pdcy_phy, !is.na(sample_data(pdcy_phy)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 1942890)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_pdc_conf[[i]] <- adonis2(as.formula(
    paste("dis ~ Group + sex + Rome_III_constip_defec_sumscore_9.15 + BMI +", i)),
    data = as(sample_data(rar), "data.frame"),
    permutations = 9999, by = "margin")["Pr(>F)"]
}

# Grab p-values into data frame
bdiv_pdc_conf_p <- data.frame(
  Var = inf_scfa_vars,
  pGroup = sapply(inf_scfa_vars, function(x) bdiv_pdc_conf[[x]][1,]),
  pVar = sapply(inf_scfa_vars, function(x) bdiv_pdc_conf[[x]][5,]))
```

Significant results:

```
kable(subset(bdiv_pdc_conf_p, pVar < 0.05),
      row.names = FALSE, booktabs = TRUE, linesep = "", digits = 3)
```

Var	pGroup	pVar
total_SCFAs	0.008	0.000
acetic_acid	0.008	0.001
propionic_acid	0.008	0.000
butyric_acid	0.016	0.000
SNGAL	0.031	0.005
Szonulin	0.008	0.002
SIL2	0.006	0.030
StoolPC	0.008	0.023

3.2.4.1 Table S3A

Collect all the results from the above comparisons into one data frame and export:

```
bdiv_combo <- data.frame(
  Variable = bdiv_all_p$Variable,
  'p (full data)' = bdiv_all_p$pVar,
  'p (control only)' = bdiv_pdc_separated$pVar_C,
  'p (PD only)' = bdiv_pdc_separated$pVar_P,
  'p (PD/C corrected)' = bdiv_pdc_p$pVar,
  'p (PD/C + confounder corrected)' = bdiv_pdc_conf_p$pVar,
  check.names = FALSE
)

# Variables with anything significant?
kable_styling(kable(bdiv_combo[rowSums(bdiv_combo[, 2:5] < 0.05) > 0,], booktabs = TRUE,
  linesep = "", digits = 3, row.names = FALSE), font_size = 8)
```

Variable	p (full data)	p (control only)	p (PD only)	p (PD/C corrected)	p (PD/C + confounder corrected)
total_SCFAs	0.000	0.000	0.066	0.000	0.000
acetic_acid	0.000	0.000	0.275	0.000	0.001
propionic_acid	0.000	0.001	0.217	0.000	0.000
butyric_acid	0.000	0.000	0.004	0.000	0.000
SNGAL	0.008	0.004	0.307	0.007	0.005
Szonulin	0.001	0.008	0.021	0.000	0.002
SIL2	0.030	0.479	0.076	0.030	0.030
StoolPC	0.016	0.598	0.054	0.022	0.023

```
# Fix variable labels
bdiv_combo$Variable <- sub(" tool", " ", varLabelFix(bdiv_combo$Variable))

# Export the full combined table
write.csv(bdiv_combo, "Outputs/tableS3A.csv", row.names = FALSE)
```

3.2.4.2 Fig S4

Plot the most interesting cases with variable values split into two categories by median for better visualization:

```
# Subsample to the same number of reads
pdcy_phy_R <- rarefy_even_depth(pdcy_phy, rngseed = 1942890)

# Calculate ordination
mds_all <- metaMDS(as.data.frame(t(as.matrix(otu_table(pdcy_phy_R)))),
  try = 500, trace = FALSE)

# Get points
mds_all_df <- data.frame(NMDS1 = mds_all$points[,1],
  NMDS2 = mds_all$points[,2],
  as(sample_data(pdcy_phy_R), "data.frame"))

# Custom plot function
bdiv_plot_2cat <- function(df, var, cols, u){
  var_bin <- factor(df[[var]] > median(df[[var]], na.rm = TRUE))
  levels(var_bin) <- substitute(c(
    paste(var, '\u2264', round(median(df[[var]], na.rm = TRUE), digits = 1), u),
    paste(var, ">", round(median(df[[var]], na.rm = TRUE), digits = 1), u)))
```

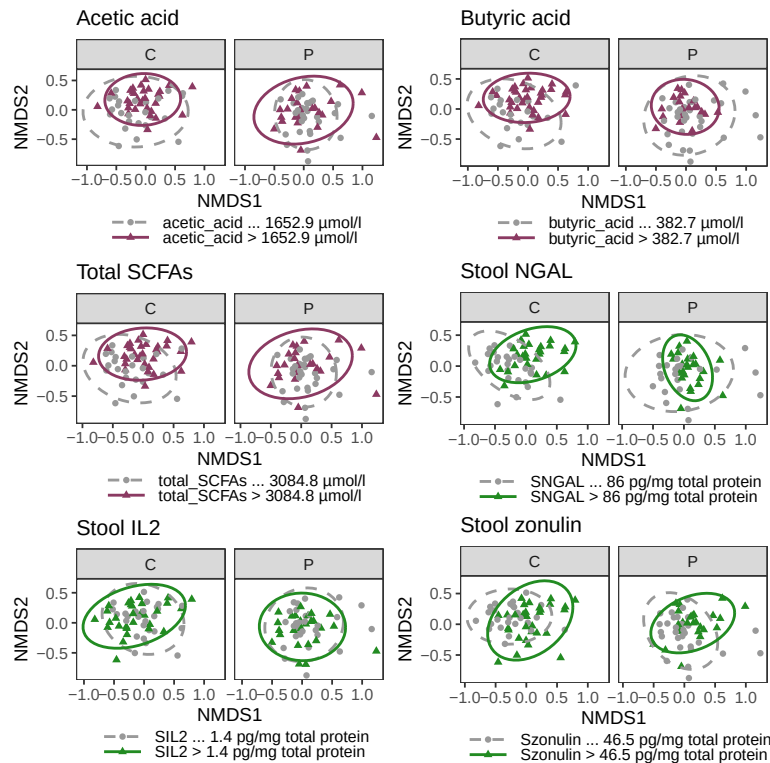
```

df_temp <- data.frame(df, var_bin = var_bin)

p <- ggplot(subset(df_temp, !is.na(var_bin)),
  aes(x = NMDS1, y = NMDS2,
    color = var_bin, linetype = var_bin, shape = var_bin)) +
  geom_point(size = 0.8) +
  theme_bw(base_size = 7) +
  coord_fixed() +
  facet_grid(~Group) +
  ggtitle(var) +
  stat_ellipse(level = 0.95, lwd = 0.5, aes(color = var_bin)) +
  scale_color_manual(values = cols, name = NULL) +
  scale_linetype_manual(values = c(2, 1), name = NULL) +
  scale_shape_manual(values = c(16, 17), name = NULL) +
  guides(color = guide_legend(nrow = 2),
    linetype = guide_legend(nrow = 2),
    shape = guide_legend(nrow = 2)) +
  theme(panel.grid = element_blank(),
    legend.position = "bottom",
    legend.margin = margin(t = -2, b = -3, unit = "mm"),
    legend.key.height = unit(2, "mm"),
    plot.margin = unit(c(-7, 1, -5, 1), "mm"),
    plot.title = element_text(size = 8))
rm(df_temp)
return(p)
}

# Draw plots
bdiv_plots <- arrangeGrob(
  bdiv_plot_2cat(mds_all_df, "acetic_acid", c("gray60", "hotpink4"),
    c('\u00B5mol/l')) + ggtitle("Acetic acid"),
  bdiv_plot_2cat(mds_all_df, "butyric_acid", c("gray60", "hotpink4"),
    c('\u00B5mol/l')) + ggtitle("Butyric acid"),
  bdiv_plot_2cat(mds_all_df, "total_SCFAs", c("gray60", "hotpink4"),
    c('\u00B5mol/l')) + ggtitle("Total SCFAs"),
  bdiv_plot_2cat(mds_all_df, "SNGAL", c("gray60", "forestgreen"),
    "pg/mg total protein") + ggtitle("Stool NGAL"),
  bdiv_plot_2cat(mds_all_df, "SIL2", c("gray60", "forestgreen"),
    "pg/mg total protein") + ggtitle("Stool IL2"),
  bdiv_plot_2cat(mds_all_df, "Szonulin", c("gray60", "forestgreen"),
    "pg/mg total protein") + ggtitle("Stool zonulin"),
  ncol = 2)
grid.arrange(bdiv_plots)

```



```
ggsave(bdiv_plots, filename = "Outputs/figS4_bdivs.pdf",
       device = cairo_pdf, width = 100, height = 120, units = "mm")
```

3.2.5 Interaction effects

Test for interactions between Group and markers/SCFAs:

```
bdiv_pdc_i <- list()

for(i in inf_scfa_vars){
  phy <- subset_samples(pdcy_phy, !is.na(sample_data(pdcy_phy)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 1942890)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_pdc_i[[i]] <- adonis2(as.formula(paste("dis ~ Group * ", i)),
                           data = as(sample_data(rar), "data.frame"),
                           permutations = 9999)["Pr(>F)"]
}

# Grab p-values into data frame
bdiv_pdc_i_p <- data.frame(
  Var = inf_scfa_vars,
  pGroup = sapply(inf_scfa_vars, function(x) bdiv_pdc_i[[x]][1,]),
  pVar = sapply(inf_scfa_vars, function(x) bdiv_pdc_i[[x]][2,]),
  pInteraction = sapply(inf_scfa_vars, function(x) bdiv_pdc_i[[x]][3,]))
```

Any significant results?

```
kable(subset(bdiv_pdc_i_p, pInteraction < 0.1),
      row.names = FALSE, booktabs = TRUE, linesep = "", digits = 3)
```

Var	pGroup	pVar	pInteraction
pIL6	0	0.409	0.035

Same, but with variables of interest split to binary by median (as in Figure S3 above):

```
# Make a df of binary variables (with new versions of the two already binary SCFA vars)
bd_meta_bin <- sapply(
  sub("_binary", "", inf_scfa_vars),
  function(x) sample_data(pdcy_phy)[[x]] >
    median(sample_data(pdcy_phy)[[x]], na.rm = TRUE))
colnames(bd_meta_bin) <- paste(colnames(bd_meta_bin), "2cat", sep = "_")

pdcy_phy_bd <- pdcy_phy
sample_data(pdcy_phy_bd) <- cbind(sample_data(pdcy_phy_bd), bd_meta_bin)

bdiv_pdc_i2 <- list()

for(i in colnames(bd_meta_bin)){
  phy <- subset_samples(pdcy_phy_bd, !is.na(sample_data(pdcy_phy_bd)[[i]]))
  rar <- rarefy_even_depth(phy, rngseed = 1942890)
  dis <- vegdist(as.data.frame(t(as.matrix(otu_table(rar)))))
  bdiv_pdc_i2[[i]] <- adonis2(as.formula(paste("dis ~ Group * ", i)),
    data = as(sample_data(rar), "data.frame"),
    permutations = 9999)["Pr(>F)"]
}

# Grab p-values into data.frame form
bdiv_pdc_i2_p <- data.frame(
  Var = colnames(bd_meta_bin),
  pGroup = sapply(colnames(bd_meta_bin), function(x) bdiv_pdc_i2[[x]][1,]),
  pVar = sapply(colnames(bd_meta_bin), function(x) bdiv_pdc_i2[[x]][2,]),
  pInteraction = sapply(colnames(bd_meta_bin), function(x) bdiv_pdc_i2[[x]][3,]))
```

Any significant results?

```
kable(subset(bdiv_pdc_i2_p, pInteraction < 0.1),
  row.names = FALSE, booktabs = TRUE, linesep = "", digits = 3)
```

Var	pGroup	pVar	pInteraction
butyric_acid_2cat	0	0.000	0.040
SNGAL_2cat	0	0.000	0.062
Szonulin_2cat	0	0.001	0.038

3.2.5.1 Table S3B

Combine and export the interaction results.

```
# Combine result data frames
bdiv_interact_res <- cbind(bdiv_pdc_i_p, bdiv_pdc_i2_p)

# Fix variable labels
bdiv_interact_res$Var <- sub(" tool", " ", varLabelFix(bdiv_interact_res$Var))

# Export
write.csv(bdiv_interact_res, "Outputs/tableS3B.csv", row.names = FALSE)
```

3.3 Enterotypes

Look for differences in levels of variables of interest in relation to [enterotypes](#). Based on the previous publication on this data, the distributions of enterotypes in the PD and control groups were slightly different, mainly with more PD subjects in the *Firmicutes* and less in the *Prevotella* enterotype. In this specific subset, the numbers were the following:

```
kable(table(pdcy_meta$Enterotype, pdcy_meta$Group), booktabs = TRUE)
```

	C	P
ET_B	8	8
ET_F	25	35
ET_P	22	10

This difference in distributions was statistically significant (Fisher's exact test $p = 0.045$).

Run tests (Kruskal-Wallis rank sum test) for significant differences in concentrations of stool and plasma markers and SCFAs between enterotypes for

1. full data
2. control subjects only
3. PD patients only,

and collect the results into one data frame:

```
# Generic testing function with Kruskal-Wallis
kwTestET <- function(df){
  res <- data.frame(Variable = inf_scfa_vars,
                    pVal = sapply(inf_scfa_vars, function(x)
                                   kruskal.test(as.formula(paste(x, "~Enterotype")), df)$p.value))

  res$pAdj <- c(
    p.adjust(res[scfa_vars, "pVal"], method = "fdr"),
    p.adjust(res[inf_vars, "pVal"], method = "fdr"),
    p.adjust(res[infp_vars, "pVal"], method = "fdr"))
  return(res)
}

# Full data
kw_et <- kwTestET(pdcy_meta)
# Only control subjects
kw_et_c <- kwTestET(subset(pdcy_meta, Group == "C"))
# Only PD
kw_et_p <- kwTestET(subset(pdcy_meta, Group == "P"))

# Combine results
kw_et_combo <- data.frame(Variable = kw_et$Variable,
                          pVal_all = kw_et$pVal,
                          pAdj_all = kw_et$pAdj,
                          pVal_C = kw_et_c$pVal,
                          pAdj_C = kw_et_c$pAdj,
                          pVal_PD = kw_et_p$pVal,
                          pAdj_PD = kw_et_p$pAdj)
```

3.3.0.1 Table S4

Significant results from the above comparisons:


```
# Variables with p < 0.05 for at least one of the three datasets
kable(kw_et_combo[rowSums(kw_et_combo[, c(2, 4, 6)] < 0.05) > 0, ],
      booktabs = TRUE, linesep = "", digits = 3, row.names = FALSE)
```

Variable	pVal_all	pAdj_all	pVal_C	pAdj_C	pVal_PD	pAdj_PD
propionic_acid	0.044	0.159	0.348	0.770	0.161	0.377
butyric_acid	0.045	0.159	0.423	0.770	0.013	0.093
SNGAL	0.001	0.008	0.002	0.008	0.419	0.936
Szonulin	0.114	0.376	0.000	0.003	0.634	0.936

```
# Fix variable names for export
kw_et_combo$Variable <- varLabelFix(kw_et_combo$Variable)

# Export the full table
write.csv(kw_et_combo, "Outputs/tableS4.csv", row.names = FALSE)
```

Run post-hoc tests (pairwise Wilcoxon rank sum test) to see which pairs of enterotypes differ:

3.3.0.1.1 NGAL

```
et_ngal_c <- as.data.frame(
  pairwise.wilcox.test(x = subset(pdcy_meta, Group == "C")$SNGAL,
    g = subset(pdcy_meta, Group == "C")$Enterotype)$p.value)
et_ngal_c$Comp <- rownames(et_ngal_c)
et_ngal_c <- melt(et_ngal_c)
colnames(et_ngal_c) <- c("ET1", "ET2", "pVal")

kable(subset(et_ngal_c, !is.na(pVal)), booktabs = TRUE, digits = 3, row.names = FALSE)
```

ET1	ET2	pVal
ET_F	ET_B	0.204
ET_P	ET_B	0.001
ET_P	ET_F	0.027

3.3.0.1.2 Zonulin

```
et_zon_c <- as.data.frame(
  pairwise.wilcox.test(x = subset(pdcy_meta, Group == "C")$Szonulin,
    g = subset(pdcy_meta, Group == "C")$Enterotype)$p.value)
et_zon_c$Comp <- rownames(et_zon_c)
et_zon_c <- melt(et_zon_c)
colnames(et_zon_c) <- c("ET1", "ET2", "pVal")

kable(subset(et_zon_c, !is.na(pVal)), booktabs = TRUE, digits = 3, row.names = FALSE)
```

ET1	ET2	pVal
ET_F	ET_B	0.374
ET_P	ET_B	0.002
ET_P	ET_F	0.002

3.3.0.1.3 Butyric acid

```
et_but_p <- as.data.frame(
  pairwise.wilcox.test(x = subset(pdcy_meta, Group == "P")$butyric_acid,
```

```

g = subset(pdcy_meta, Group == "P")$Enterotype)$p.value)
et_but_p$Comp <- rownames(et_but_p)
et_but_p <- melt(et_but_p)
colnames(et_but_p) <- c("ET1", "ET2", "pVal")
kable(subset(et_but_p, !is.na(pVal)), booktabs = TRUE, digits = 3, row.names = FALSE)

```

ET1	ET2	pVal
ET_F	ET_B	0.939
ET_P	ET_B	0.017
ET_P	ET_F	0.017

3.3.0.2 Fig S5

Combine the results, make a plot and export:

```

# Handmade DF for p-value annotations
et_pvals <- rbind(
  data.frame(subset(et_but_p, pVal < 0.05),
    Group = "P",
    variable = "butyric_acid"),
  data.frame(subset(et_ngal_c, pVal < 0.05),
    Group = "C",
    variable = "SNGAL"),
  data.frame(subset(et_zon_c, pVal < 0.05),
    Group = "C",
    variable = "Szonulin"))
et_pvals$stars <- stars.pval(et_pvals$pVal)
et_pvals$xttext <- c(2, 2.5)
et_pvals$x1 <- c(1, 2)
et_pvals$y <- c(2.6, 3.1, 325, 375, 190, 225)
et_pvals$ytext <- c( 2.7, 3.2, 330, 380, 195, 230)

et_df <- melt(pdcy_meta[,c("Group", "Enterotype", "butyric_acid", "SNGAL", "Szonulin")])
et_df <- subset(et_df, !is.na(Enterotype))
et_df[et_df$variable == "butyric_acid", "value"] <-
  et_df[et_df$variable == "butyric_acid", "value"]/1000

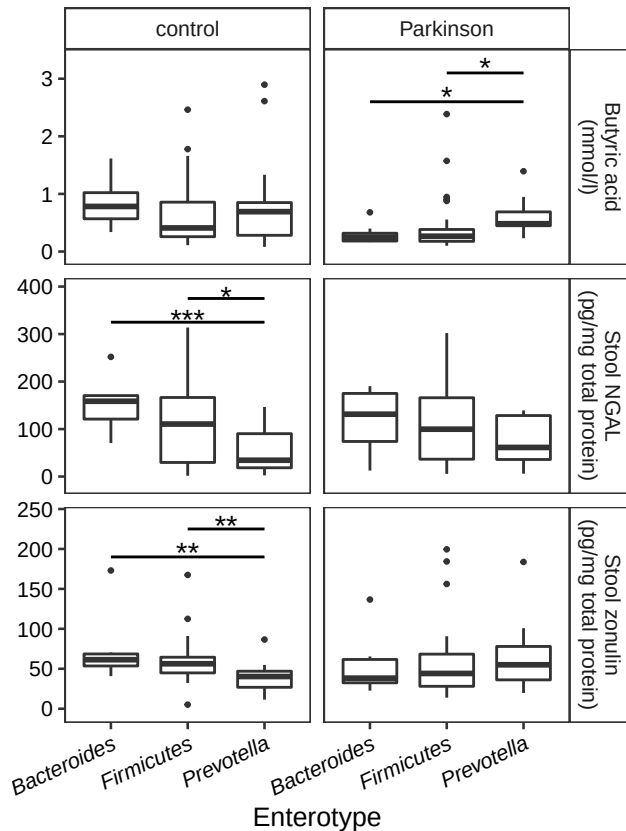
et_plots <- ggplot(et_df, aes(x = Enterotype, y = value)) +
  geom_boxplot(width = 0.7, outlier.size = 0.5) +
  theme_bw(base_size = 10) +
  xlab("Enterotype") +
  ylab(NULL) +
  scale_x_discrete(labels = c("Bacteroides", "Firmicutes", "Prevotella")) +
  scale_y_continuous(expand = expansion(mult = 0.1)) +
  facet_grid(variable ~ Group, scales = "free",
    labeller = labeller(
      variable = setNames(c("Butyric acid\n(mmol/l)",
        "Stool NGAL\n(pg/mg total protein)",
        "Stool zonulin\n(pg/mg total protein)"),
        c("butyric_acid", "SNGAL", "Szonulin")),
      Group = pdc_labels)) +
  geom_text(aes(x = xttext, y = ytext, label = stars),
    et_pvals, inherit.aes = FALSE, size = 4) +
  geom_segment(aes(x = x1, xend = 3, y = y, yend = y),

```

```

    et_pvals, inherit.aes = FALSE) +
  theme(panel.grid = element_blank(),
        axis.text = element_text(color = "black"),
        strip.background = element_rect(fill = "white"),
        axis.text.x = element_text(angle = 25, vjust = 1, hjust = 1, face = "italic"))
et_plots

```



```

ggsave(et_plots, filename = "Outputs/figS5_enterotypes.pdf",
       width = maxwd/2, height = maxhi/2, units = "mm")

```

3.4 Differential abundance with DESeq2

Look for bacterial genera and families associated with specific markers/SCFAs with DESeq2, focusing on variables that were particularly interesting based on the other comparisons: stool calprotectin, NGAL, zonulin, and the three most common SCFAs. Run the comparisons in four different ways:

1. Taxon ~ variable, full data
2. Taxon ~ variable + sex + Rome III 9-15 score + PD/C, full data
3. Taxon ~ variable, control subjects only
4. Taxon ~ variable, PD patients only

Set up data and functions for testing:

```

# Functions

# One variable, Taxon ~ variable of interest
ds2_1var <- function(phyloObj, v){
  phyloObj <- prune_samples(as.vector(!is.na(sample_data(phyloObj)[,v])), phyloObj)
  ds <- phyloseq_to_deseq2(phyloObj, as.formula(paste("~", v)))
}

```

```

ds <- DESeq(ds, fitType = "parametric", sfType = "poscounts")
dsres <- cbind(data.frame(results(ds)), as(tax_table(phyloObj), "matrix"))

return(dsres)
}

# Confounder-corrected:
# Taxon ~ variable of interest + sex + Rome III score + PD/control
# (grabbing results for the variable of interest only!)
ds2_lvar_pdc <- function(phyloObj, v){
  phyloObj <- prune_samples(as.vector(!is.na(sample_data(phyloObj)[,v])), phyloObj)
  ds <- phyloseq_to_deseq2(phyloObj, as.formula(
    paste("~", v, "+ sex + Rome_III_constip_defec_sumscore_9.15 + Group")))
  ds <- DESeq(ds, fitType = "parametric", sfType = "poscounts")
  dsres <- cbind(data.frame(results(ds, name = v)),
    as(tax_table(phyloObj), "matrix"))

  return(dsres)
}

# Phyloseq objects

# Function for trimming rare taxa and unclassifieds from the phyloseq objects
ds2trim <- function(phyloIn){
  lv <- tail(colnames(tax_table(phyloIn)), n = 1)
  phyloIn <- filter_taxa(phyloIn, function(x) sum(x > 5) > nsamples(pdcy_phy)/2,
    prune = TRUE)
  prune_taxa(as.vector(tax_table(phyloIn)[, lv] != "unclassified"), phyloIn)
}

# Full data
pdcy_phy_gen_trim <- ds2trim(pdcy_phy_gen)
pdcy_phy_fam_trim <- ds2trim(pdcy_phy_fam)

# Make the PD-only and C-only subsets by subsetting these objects,
# so that all comparisons have the same set of taxa:

# Controls only
pdcy_phy_C_gen_trim <- subset_samples(pdcy_phy_gen_trim, Group == "C")
pdcy_phy_C_fam_trim <- subset_samples(pdcy_phy_fam_trim, Group == "C")

# PD only
pdcy_phy_P_gen_trim <- subset_samples(pdcy_phy_gen_trim, Group == "P")
pdcy_phy_P_fam_trim <- subset_samples(pdcy_phy_fam_trim, Group == "P")

```

3.4.1 SCFAs

Run comparisons for SCFAs:

```

# Variables of interest
da_scfa_vars <- c("acetic_acid", "butyric_acid", "propionic_acid")

# Family level comparisons
fam_scfa <- do.call("rbind",

```

```

c(lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var(pdcy_phy_fam_trim, x),
                                SCFA = x,
                                Subset = "all")),
  lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var_pdc(pdcy_phy_fam_trim, x),
                                SCFA = x,
                                Subset = "pdc")),
  lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var(pdcy_phy_C_fam_trim, x),
                                SCFA = x,
                                Subset = "C")),
  lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var(pdcy_phy_P_fam_trim, x),
                                SCFA = x,
                                Subset = "P"))))

# Genus level comparisons
gen_scfa <- do.call("rbind",
  c(lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var(pdcy_phy_gen_trim, x),
                                SCFA = x,
                                Subset = "all")),
    lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var_pdc(pdcy_phy_gen_trim, x),
                                SCFA = x,
                                Subset = "pdc")),
    lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var(pdcy_phy_C_gen_trim, x),
                                SCFA = x,
                                Subset = "C")),
    lapply(da_scfa_vars,
        function(x) data.frame(ds2_1var(pdcy_phy_P_gen_trim, x),
                                SCFA = x,
                                Subset = "P"))))

```

Collect and reorganize results:

```

# List taxa significant in any comparison
fam_scfa_res_df <- subset(fam_scfa,
                        Family %in% unique(subset(fam_scfa, padj < 0.05)$Family))
fam_scfa_res_df$Taxon <- fam_scfa_res_df$Family
fam_scfa_res_df$Genus <- NA

gen_scfa_res_df <- subset(gen_scfa,
                        Genus %in% unique(subset(gen_scfa, padj < 0.05)$Genus))
gen_scfa_res_df$Taxon <- gen_scfa_res_df$Genus

# Combined data frame
scfa_res_df <- rbind(data.frame(fam_scfa_res_df, Level = "Family"),
                    data.frame(gen_scfa_res_df, Level = "Genus"))

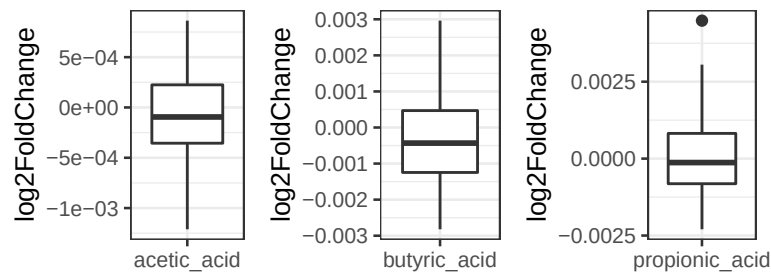
# Reorganize for plotting
scfa_res_df_m <- melt(scfa_res_df)

```

```
scfa_res_df_c <- dcast(scfa_res_df_m, SCFA + Taxon + Subset + Level ~ variable)
scfa_res_df_c$pStar <- stars.pval(scfa_res_df_c$padj)
```

Check for outlierish log2 fold changes:

```
# Check the spreads of values for the log2 fc
do.call("grid.arrange",
  c(lapply(da_scfa_vars, function(x)
    ggplot(subset(scfa_res_df_c, SCFA == x),
      aes(x = SCFA, y = log2FoldChange)) +
    geom_boxplot() +
    xlab(NULL) +
    theme_bw(base_size = 10)), nrow = 1))
```



No major outliers. Export the final table:

3.4.1.1 Table S5A

```
# Rename the group levels
scfa_res_df_c$Subset <- factor(scfa_res_df_c$Subset, levels = c("all", "pdc", "C", "P"))
levels(scfa_res_df_c$Subset) <- c("full data",
  "full data, corrected for PD/C and confounders",
  "control only", "PD only")

# Nicer names for the SCFA var
scfa_res_df_c$SCFA <- capitalize(sub("_", " ", sub("", "", scfa_res_df_c$SCFA)))

# Export full results
write.csv(scfa_res_df_c, "Outputs/tableS5A.csv", row.names = FALSE)
```

3.4.1.2 Fig4: SCFAs

Summary plot of the results:

```
# Plotting function
inf_heatmap <- function(df, v, locol, hicol){
  ggplot(subset(df, df[, 1] == v),
    aes(x = Subset, y = Taxon, fill = log2FoldChange)) +
  geom_tile(color = "gray30") +
  ggtitle(v) +
  theme_bw(base_size = 9) +
  scale_fill_gradient2(low = locol, mid = "white", high = hicol,
    name = "log2\nfold\nchange") +
  geom_text(aes(label = pStar), size = 5) +
  theme(legend.position = "bottom",
    axis.title = element_blank(),
    panel.grid = element_blank(),
    axis.text.y = element_blank(),
```

```

    axis.text.x = element_text(color = "black"),
    axis.ticks = element_blank(),
    legend.key.width = unit(0.55, "cm"),
    legend.title = element_blank()
}

# Genus-level df for plotting
scfa_res_df_c_gen <- subset(scfa_res_df_c, Level == "Genus")

# Rename the group levels for plotting
levels(scfa_res_df_c_gen$Subset) <- c("full\\ndata",
    "full\\ndata,\\ncorrected\\nfor PD/C and\\nconfounders",
    "control\\nonly", "PD\\nonly")

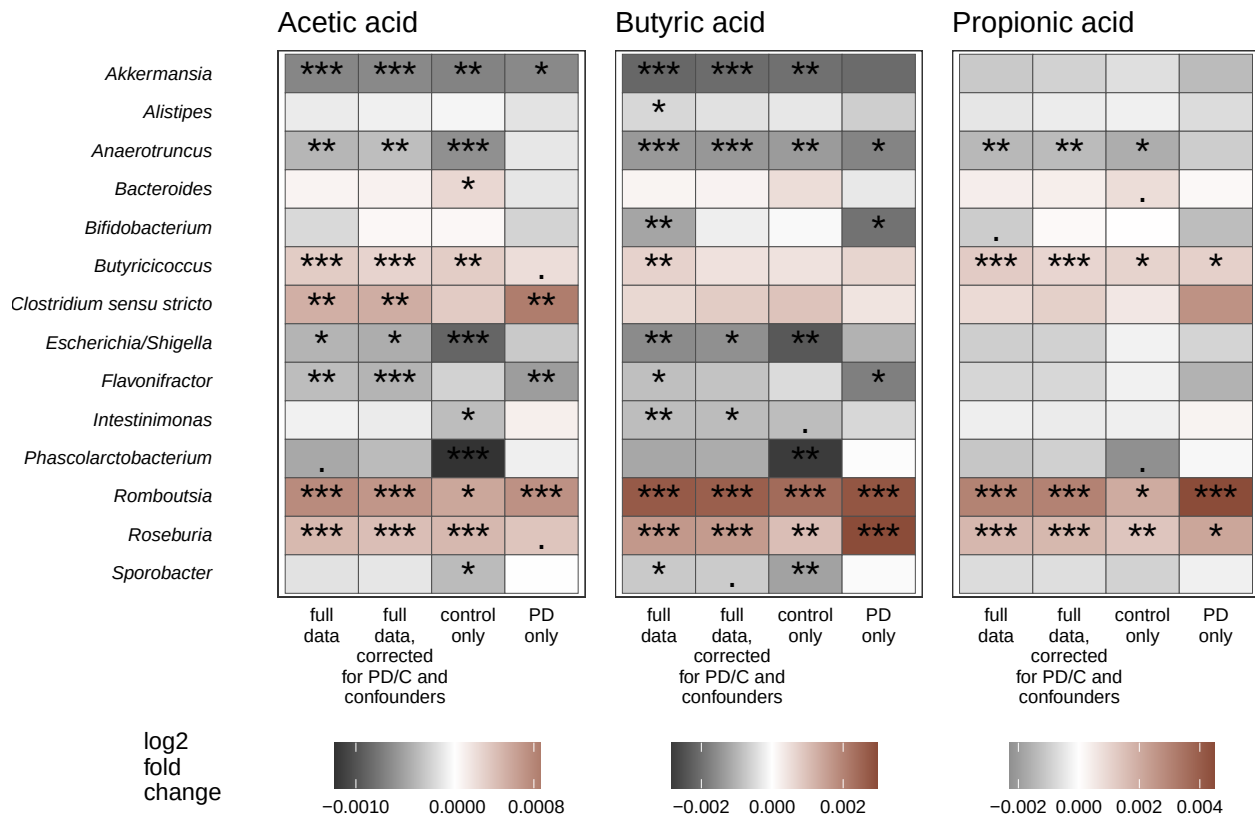
# Nicer names & reordering for the taxon list
scfa_res_df_c_gen$Taxon <- gsub("_", " ", scfa_res_df_c_gen$Taxon)
scfa_res_df_c_gen$Taxon <- factor(scfa_res_df_c_gen$Taxon,
    levels = rev(sort(unique(scfa_res_df_c_gen$Taxon))))

# Workaround to create y-axis labels + legend title
scfa_y <- ggplot(subset(scfa_res_df_c_gen,
    Subset == levels(scfa_res_df_c_gen$Subset)[2]),
    aes(x = Subset, y = Taxon, fill = log2FoldChange)) +
  geom_tile(color = "white", width = 0) +
  theme_bw(base_size = 9) +
  ggtitle(" ") +
  scale_fill_gradient2(low = "white", mid = "white", high = "white",
    name = "log2\\nfold\\nchange") +
  theme(legend.position = "bottom",
    legend.key.size = unit(0, "mm"),
    legend.text = element_blank(),
    legend.margin = margin(r = 30, b = 10),
    axis.title = element_blank(),
    panel.grid = element_blank(),
    panel.background = element_rect(color = "white"),
    panel.border = element_rect(color = "white"),
    axis.text.x = element_text(color = "white"),
    axis.text.y = element_text(color = "black", face = "italic",
      margin = margin(r = -4)),
    axis.ticks = element_blank())

# Put together plots
da_scfa_plots <- arrangeGrob(
  scfa_y,
  inf_heatmap(scfa_res_df_c_gen, "Acetic acid", "gray25", "salmon4") +
    scale_fill_gradient2(low = "gray20", mid = "white", high = "salmon4",
      name = "log2\\nfold\\nchange", breaks = c(-0.001, 0, 0.0008),
      labels = function(x) format(x, scientific = FALSE)),
  inf_heatmap(scfa_res_df_c_gen, "Butyric acid", "gray20", "salmon4") +
    scale_fill_gradient2(low = "gray20", mid = "white", high = "salmon4",
      name = "log2\\nfold\\nchange", breaks = c(-0.002, 0, 0.002)),
  inf_heatmap(scfa_res_df_c_gen, "Propionic acid", "gray20", "salmon4"),
  nrow = 1, widths = c(1.55, 2, 2, 2))

```

```
grid.arrange(da_scfa_plots)
```



3.4.1.3 Fig S6

Scatterplots of the most interesting taxa from the results.

```
# Set up a relative abundance data.frame
ra_df_gen <- t(prop.table(as(otu_table(pdcy_phy_gen), "matrix"), 2)*100)
ra_df <- cbind(ra_df_gen, sample_data(pdcy_phy_gen)[, c("Group", inf_scfa_vars)])

# Remove the doubled "unclassified" column
ra_df <- ra_df[, -(grep("unclassified", colnames(ra_df)))]

# Scatterplot function for relative abundances
ra_scatter <- function(df, xvar, taxon){
  ggplot(df, aes_string(x = xvar, y = taxon)) +
    geom_point(size = 0.75) +
    theme_bw(base_size = 8) +
    facet_grid(~Group) +
    ylab("Relative abundance (%)") +
    ggtitle(taxon) +
    theme(panel.grid = element_blank(),
          axis.text = element_text(color = "black"),
          plot.title = element_text(face = "italic"))
}
```

Since the three SCFAs had overall similar patterns for the taxa, only plot each taxon once:


```

# Significant taxa for butyric acid
buty_sigs <- as.character(unique(
  subset(scfa_res_df_c_gen, SCFA == "Butyric acid" & padj < 0.1)$Taxon))

# Significant taxa for acetic acid that are not on the butyric acid list
acet_sigs <- setdiff(as.character(unique(
  subset(scfa_res_df_c_gen, SCFA == "Acetic acid" & padj < 0.1)$Taxon)), buty_sigs)

# Any significant taxa for propionic acid not on the butyric acid or acetic acid list?
setdiff(setdiff(as.character(unique(
  subset(scfa_res_df_c_gen, SCFA == "Propionic acid" & padj < 0.1)$Taxon)),
  buty_sigs), acet_sigs)

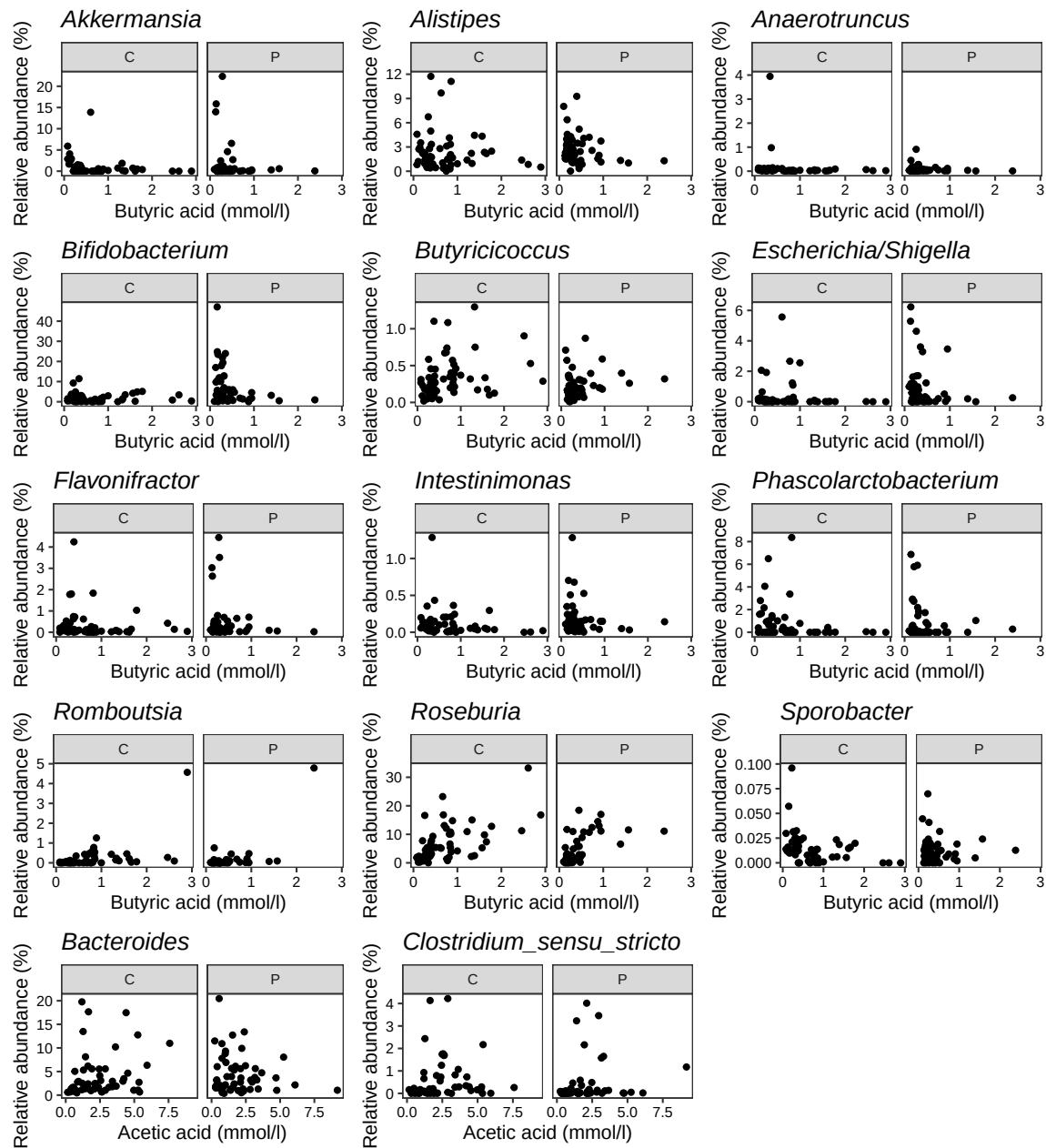
## character(0)

# Fix the problematically named Escherichia/Shigella
buty_sigs[grep("Escher", buty_sigs)] <- "`Escherichia/Shigella`"
# Fix the spaces in Clostridium sensu stricto
acet_sigs[grep("Clost", acet_sigs)] <- gsub(" ", "_", acet_sigs[grep("Clost", acet_sigs)])

# Plot
scfa_da_scatters <- c(
  lapply(buty_sigs, function(x)
    ra_scatter(ra_df, "butyric_acid/1000", x) +
    xlab("Butyric acid (mmol/l)")),
  lapply(acet_sigs, function(x)
    ra_scatter(ra_df, "acetic_acid/1000", x) +
    xlab("Acetic acid (mmol/l)")))
# Fix the title for 'Escherichia/Shigella'
scfa_da_scatters[[grep("Escher", buty_sigs)]] <-
  scfa_da_scatters[[grep("Escher", buty_sigs)]] +
  ggtitle("Escherichia/Shigella")

# Arrange plots
scfa_da_scatters <- do.call(arrangeGrob, c(scfa_da_scatters, ncol = 3))
grid.arrange(scfa_da_scatters)

```



```
# Export
ggsave(scfa_da_scatters,
       filename = "Outputs/figS6_diffabund_scatters_scfas.pdf",
       width = maxwd, height = maxhi*0.9, units = "mm")
```

3.4.2 Inflammatory markers

Run comparisons for stool markers:

```
# Variables of interest
da_inf_vars <- c("SNGAL", "Szonulin", "Scalprotectin")

# Family level comparisons
fam_inf <- do.call("rbind",
                  c(lapply(da_inf_vars,
```

```

        function(x) data.frame(ds2_1var(pdcy_phy_fam_trim, x),
                               Marker = x,
                               Subset = "all")),
lapply(da_inf_vars,
       function(x) data.frame(ds2_1var_pdc(pdcy_phy_fam_trim, x),
                               Marker = x,
                               Subset = "pdc")),
lapply(da_inf_vars,
       function(x) data.frame(ds2_1var(pdcy_phy_C_fam_trim, x),
                               Marker = x,
                               Subset = "C")),
lapply(da_inf_vars,
       function(x) data.frame(ds2_1var(pdcy_phy_P_fam_trim, x),
                               Marker = x,
                               Subset = "P"))))

# Genus level comparisons:
gen_inf <- do.call("rbind",
                  c(lapply(da_inf_vars,
                           function(x) data.frame(ds2_1var(pdcy_phy_gen_trim, x),
                                                       Marker = x,
                                                       Subset = "all")),
                    lapply(da_inf_vars,
                           function(x) data.frame(ds2_1var_pdc(pdcy_phy_gen_trim, x),
                                                       Marker = x,
                                                       Subset = "pdc")),
                    lapply(da_inf_vars,
                           function(x) data.frame(ds2_1var(pdcy_phy_C_gen_trim, x),
                                                       Marker = x,
                                                       Subset = "C")),
                    lapply(da_inf_vars,
                           function(x) data.frame(ds2_1var(pdcy_phy_P_gen_trim, x),
                                                       Marker = x,
                                                       Subset = "P"))))

```

Collect and reorganize results:

```

# List taxa significant in any comparison
fam_inf_res_df <- subset(fam_inf, Family %in% unique(subset(fam_inf, padj < 0.05)$Family))
fam_inf_res_df$Taxon <- fam_inf_res_df$Family
fam_inf_res_df$Genus <- NA

gen_inf_res_df <- subset(gen_inf, Genus %in% unique(subset(gen_inf, padj < 0.05)$Genus))
gen_inf_res_df$Taxon <- gen_inf_res_df$Genus

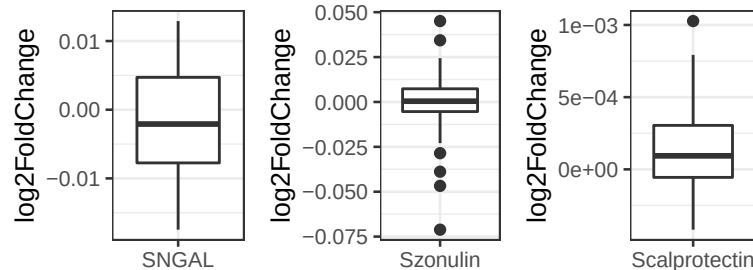
# Combined data frame
inf_res_df <- rbind(data.frame(fam_inf_res_df, Level = "Family"),
                   data.frame(gen_inf_res_df, Level = "Genus"))

# Reorganize for plotting
inf_res_df_m <- melt(inf_res_df)
inf_res_df_c <- dcast(inf_res_df_m, Marker + Taxon + Subset + Level ~ variable)
inf_res_df_c$padjStar <- stars.pval(inf_res_df_c$padj)

```

Check the spreads of values for the log2 fold changes:

```
# Check the spreads of values for the log2 fc
do.call("grid.arrange",
      c(lapply(da_inf_vars, function(x)
        ggplot(subset(inf_res_df_c, Marker == x),
          aes(x = Marker, y = log2FoldChange)) +
        geom_boxplot() +
        xlab(NULL) +
        theme_bw(base_size = 10)), nrow = 1))
```



No major outliers. Export and plot the final results.

3.4.2.1 Table S5B

```
# Rename the group levels
inf_res_df_c$Subset <- factor(inf_res_df_c$Subset, levels = c("all", "pdc", "C", "P"))
levels(inf_res_df_c$Subset) <- c("full data",
  "full data, corrected for PD/C and confounders",
  "control only", "PD only")

# Nicer names for the Marker variable
inf_res_df_c$Marker <- sub("^S", "Stool ", inf_res_df_c$Marker)

# Export full results
write.csv(inf_res_df_c, "Outputs/tableS5B.csv", row.names = FALSE)
```

3.4.2.2 Fig 4: markers

```
# Add line breaks to group levels
levels(inf_res_df_c$Subset) <- c("full\\ndata",
  "full\\ndata,\\ncorrected\\nfor PD/C and\\nconfounders",
  "control\\nonly", "PD\\nonly")

# Reverse taxon list to have the alphabetical order from top to bottom
inf_res_df_c$Taxon <- factor(inf_res_df_c$Taxon,
  levels = rev(sort(levels(inf_res_df_c$Taxon))))

# Trim to genera for plot
inf_res_df_c_gen <- subset(inf_res_df_c, Level == "Genus")

# Italic labels as an expression
da_marker_taxa <- rev(unique(subset(inf_res_df_c_gen,
  Subset == levels(inf_res_df_c$Subset)[2])$Taxon))

da_marker_tax_labels <- parse(text = c(
  paste("italic('", gsub(".*:", "", da_marker_taxa), "')", sep = "")))
```

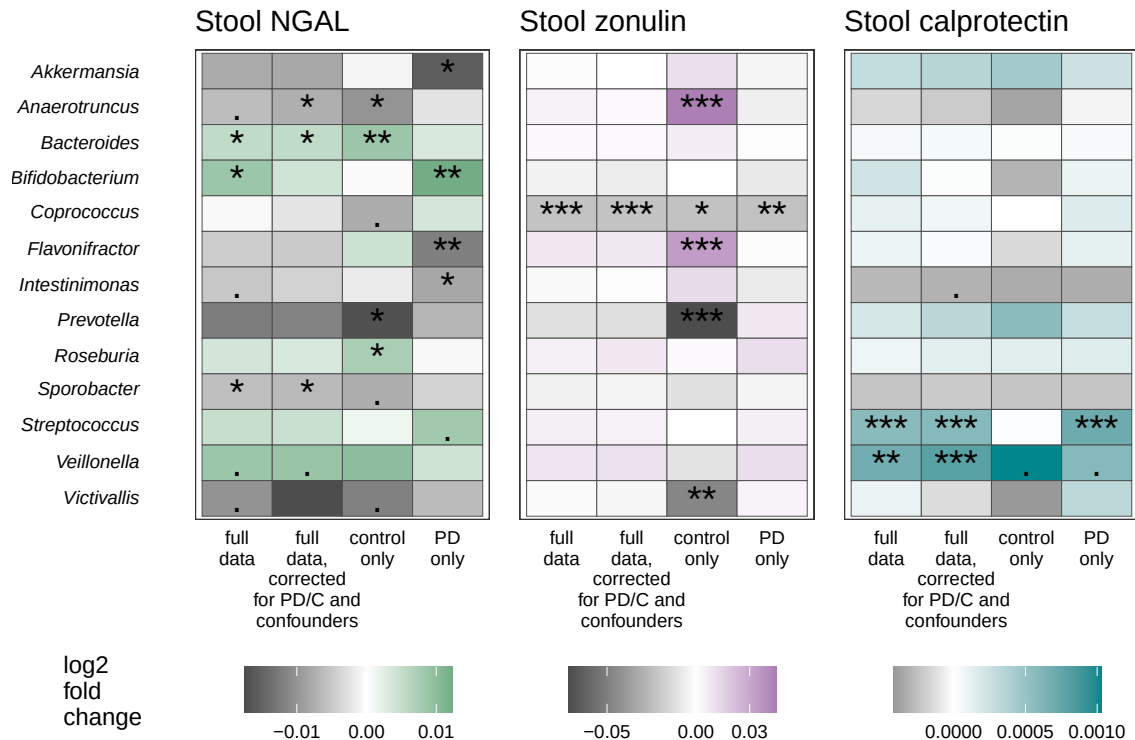
```

# Workaround to create only y-axis labels + legend title
marker_y <- ggplot(subset(inf_res_df_c_gen,
                          Subset == levels(inf_res_df_c$Subset)[2]),
                  aes(x = Subset, y = Taxon, fill = log2FoldChange)) +
  geom_tile(color = "white", width = 0) +
  theme_bw(base_size = 9) +
  ggtitle(" ") +
  scale_fill_gradient2(low = "white", mid = "white", high = "white",
                      name = "log2\nfold\nchange") +
  scale_y_discrete(labels = da_marker_tax_labels) +
  theme(legend.position = "bottom",
        legend.key.size = unit(0, "mm"),
        legend.text = element_blank(),
        legend.margin = margin(r = 30, b = 10),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        panel.background = element_rect(color = "white"),
        panel.border = element_rect(color = "white"),
        axis.text.x = element_text(color = "white"),
        axis.text.y = element_text(color = "black", margin = margin(r = -5, "mm")),
        axis.ticks = element_blank())

# Put together plots
da_marker_plots <- arrangeGrob(
  marker_y,
  inf_heatmap(inf_res_df_c_gen, "Stool NGAL", "gray30", "seagreen4"),
  inf_heatmap(inf_res_df_c_gen, "Stool zonulin", "gray30", "mediumorchid4") +
    scale_fill_gradient2(low = "gray30", mid = "white", high = "mediumorchid4",
                        name = "log2\nfold\nchange", breaks = c(-0.05, 0, 0.03)),
  inf_heatmap(inf_res_df_c_gen, "Stool calprotectin", "gray10", "turquoise4") +
    scale_fill_gradient2(low = "gray10", mid = "white", high = "turquoise4",
                        name = "log2\nfold\nchange", breaks = c(0, 0.0005, 0.0010),
                        labels = function(x) format(x, scientific = FALSE)),
  nrow = 1, widths = c(1.1, 2, 2, 2))

grid.arrange(da_marker_plots)

```



3.4.2.3 Fig 4

Combine and export the plots for differentially abundant genera for SCFAs and markers:

```
# Blank panel for plotting
library("grid")
blankPanel <- grid.rect(gp = gpar(col = "white"), draw = FALSE)

da_plots <- arrangeGrob(da_scfa_plots,
  arrangeGrob(blankPanel, da_marker_plots, nrow = 1, widths = c(0.08, 1)),
  heights = c(length(unique(scfa_res_df_c_gen$Taxon)),
    length(unique(inf_res_df_c_gen$Taxon))))

# Export combined figure
bitmap("Outputs/fig4_da_plots.tif", width = 140, height = 210,
  units = "mm", res = 600)
grid.arrange(da_plots)
dev.off()

## pdf
## 2
```

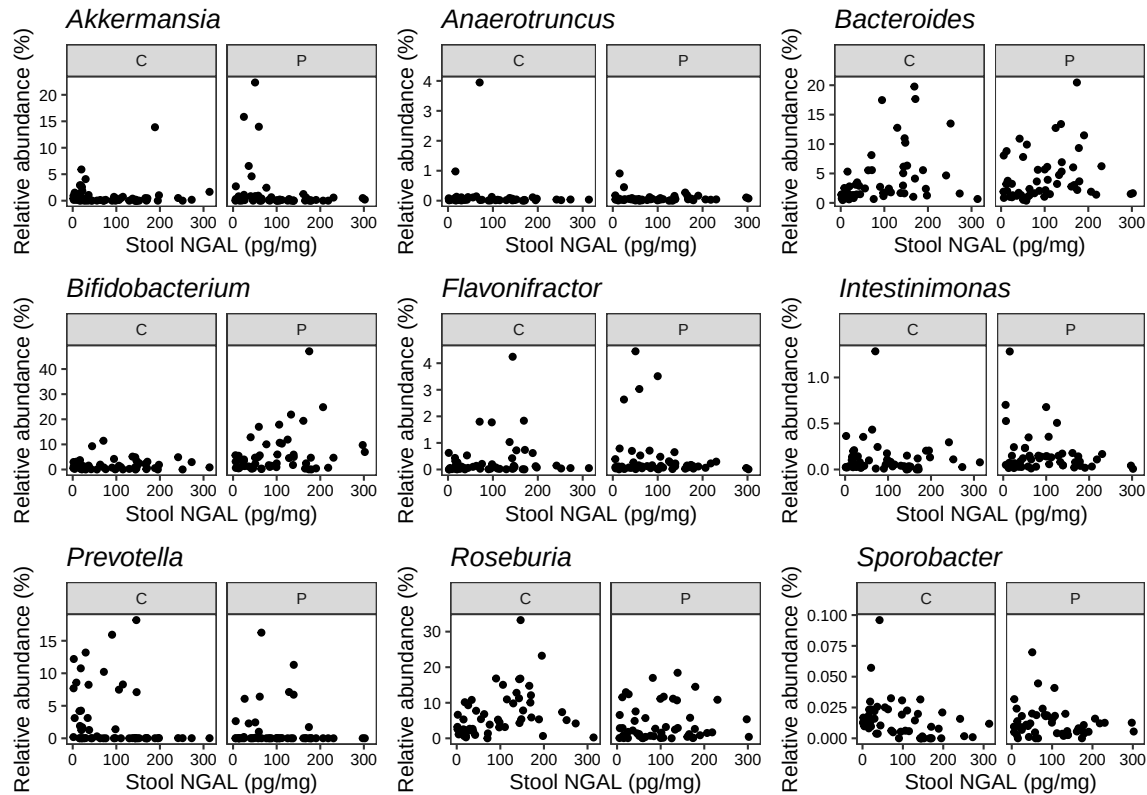
3.4.2.4 Fig S7

Separate plots for the most interesting taxa for each stool inflammatory marker:

NGAL

```
# List of significant taxa
ngal_sigs <- as.character(unique(
  subset(inf_res_df_c_gen, Marker == "Stool NGAL" & padj < 0.05)$Taxon))
```

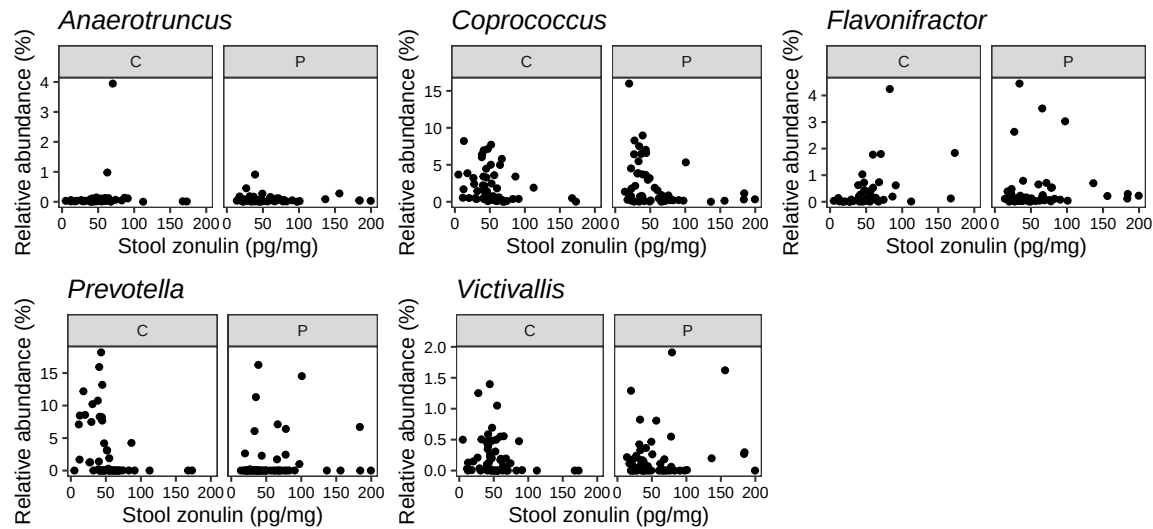
```
# Plot
ngal_da_scatters <- do.call(
  arrangeGrob, c(lapply(ngal_sigs, function(x)
    ra_scatter(ra_df, "SNGAL", x) +
    xlab("Stool NGAL (pg/mg)")), ncol = 3))
grid.arrange(ngal_da_scatters)
```



Zonulin

```
zonu_sigs <- as.character(unique(
  subset(inf_res_df_c_gen, Marker == "Stool zonulin" & padj < 0.05)$Taxon))

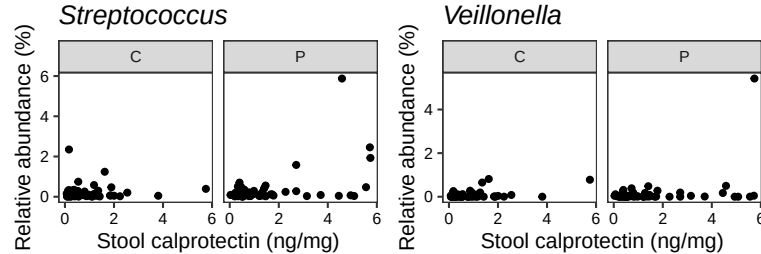
zonu_da_scatters <- do.call(
  arrangeGrob, c(lapply(zonu_sigs, function(x)
    ra_scatter(ra_df, "Szonulin", x) +
    xlab("Stool zonulin (pg/mg)")), ncol = 3))
grid.arrange(zonu_da_scatters)
```



Calprotectin

```
calp_sigs <- as.character(unique(
  subset(inf_res_df_c_gen, Marker == "Stool calprotectin" & padj < 0.05)$Taxon))

calp_da_scatters <- do.call(
  arrangeGrob, c(lapply(calp_sigs, function(x)
    ra_scatter(ra_df, "Scalprotectin/1000", x) +
    xlab("Stool calprotectin (ng/mg)")), ncol = 3))
grid.arrange(calp_da_scatters)
```



Export the figures together:

```
library("grid")
# (repeated because for some reason knitr seems to forget it?)

pdf("Outputs/figS7_diffabund_scatters_markers.pdf",
  width = 7.5, height = 10)
grid.arrange(
  textGrob("A", x = unit(0.025, "npc"), y = unit(0.4, "npc"),
    gp = gpar(fontface = "bold", fontsize = 12)),
  ngal_da_scatters,
  textGrob("B", x = unit(0.025, "npc"), y = unit(0.4, "npc"),
    gp = gpar(fontface = "bold", fontsize = 12)),
  zonu_da_scatters,
  textGrob("C", x = unit(0.025, "npc"), y = unit(0.4, "npc"),
    gp = gpar(fontface = "bold", fontsize = 12)),
  calp_da_scatters,
  ncol = 1, heights = c(0.15, 3, 0.15, 2, 0.15, 1))
dev.off()
```



```
## pdf
## 2
```

4 Session info

List all R packages used for these analyses:

```
devtools::session_info()
```

```
## - Session info -----
## setting value
## version R version 3.6.2 (2019-12-12)
## os      macOS Catalina 10.15.4
## system  x86_64, darwin15.6.0
## ui      X11
## language (EN)
## collate en_US.UTF-8
## ctype   en_US.UTF-8
## tz      Europe/Luxembourg
## date    2020-05-29
##
## - Packages -----
## package      * version date      lib source
## acepack      1.4.1   2016-10-29 [1] CRAN (R 3.6.0)
## ade4         1.7-15  2020-02-13 [1] CRAN (R 3.6.0)
## annotate     1.64.0  2019-10-29 [1] Bioconductor
## AnnotationDbi 1.48.0  2019-10-29 [1] Bioconductor
## ape          5.3     2019-03-17 [1] CRAN (R 3.6.0)
## assertthat   0.2.1   2019-03-21 [1] CRAN (R 3.6.0)
## backports    1.1.6   2020-04-05 [1] CRAN (R 3.6.2)
## base64enc    0.1-3   2015-07-28 [1] CRAN (R 3.6.0)
## Biobase      * 2.46.0  2019-10-29 [1] Bioconductor
## BiocGenerics * 0.32.0  2019-10-29 [1] Bioconductor
## BiocParallel * 1.20.1  2019-12-21 [1] Bioconductor
## biomformat   1.14.0  2019-10-29 [1] Bioconductor
## Biostrings   2.54.0  2019-10-29 [1] Bioconductor
## bit          1.1-15.2 2020-02-10 [1] CRAN (R 3.6.0)
## bit64        0.9-7   2017-05-08 [1] CRAN (R 3.6.0)
## bitops       1.0-6   2013-08-17 [1] CRAN (R 3.6.0)
## blob         1.2.1   2020-01-20 [1] CRAN (R 3.6.0)
## broom        0.5.6   2020-04-20 [1] CRAN (R 3.6.2)
## callr        3.4.3   2020-03-28 [1] CRAN (R 3.6.2)
## checkmate    2.0.0   2020-02-06 [1] CRAN (R 3.6.0)
## cli          2.0.2   2020-02-28 [1] CRAN (R 3.6.0)
## cluster      2.1.0   2019-06-19 [1] CRAN (R 3.6.2)
## codetools    0.2-16  2018-12-24 [1] CRAN (R 3.6.2)
## colorspace   1.4-1   2019-03-18 [1] CRAN (R 3.6.0)
## crayon       1.3.4   2017-09-16 [1] CRAN (R 3.6.0)
## data.table   1.12.8  2019-12-09 [1] CRAN (R 3.6.0)
## DBI          1.1.0   2019-12-15 [1] CRAN (R 3.6.0)
## DelayedArray * 0.12.3  2020-04-09 [1] Bioconductor
## desc         1.2.0   2018-05-01 [1] CRAN (R 3.6.0)
## DESeq2       * 1.26.0  2019-10-29 [1] Bioconductor
```

##	devtools	2.3.0	2020-04-10	[1]	CRAN	(R 3.6.2)
##	digest	0.6.25	2020-02-23	[1]	CRAN	(R 3.6.0)
##	dplyr	0.8.5	2020-03-07	[1]	CRAN	(R 3.6.0)
##	ellipsis	0.3.0	2019-09-20	[1]	CRAN	(R 3.6.0)
##	evaluate	0.14	2019-05-28	[1]	CRAN	(R 3.6.0)
##	factoextra	* 1.0.7	2020-04-01	[1]	CRAN	(R 3.6.2)
##	fansi	0.4.1	2020-01-08	[1]	CRAN	(R 3.6.0)
##	farver	2.0.3	2020-01-16	[1]	CRAN	(R 3.6.0)
##	foreach	1.5.0	2020-03-30	[1]	CRAN	(R 3.6.2)
##	foreign	0.8-76	2020-03-03	[1]	CRAN	(R 3.6.0)
##	Formula	* 1.2-3	2018-05-03	[1]	CRAN	(R 3.6.0)
##	fs	1.4.1	2020-04-04	[1]	CRAN	(R 3.6.2)
##	genefilter	1.68.0	2019-10-29	[1]	Bioconductor	
##	geneplotter	1.64.0	2019-10-29	[1]	Bioconductor	
##	generics	0.0.2	2018-11-29	[1]	CRAN	(R 3.6.0)
##	GenomeInfoDb	* 1.22.1	2020-03-27	[1]	Bioconductor	
##	GenomeInfoDbData	1.2.2	2020-03-09	[1]	Bioconductor	
##	GenomicRanges	* 1.38.0	2019-10-29	[1]	Bioconductor	
##	GGally	* 1.5.0	2020-03-25	[1]	CRAN	(R 3.6.0)
##	ggcorrplot	* 0.1.3	2019-05-19	[1]	CRAN	(R 3.6.0)
##	ggfortify	* 0.4.9	2020-03-11	[1]	CRAN	(R 3.6.0)
##	ggplot2	* 3.3.0	2020-03-05	[1]	CRAN	(R 3.6.0)
##	ggpubr	0.2.5	2020-02-13	[1]	CRAN	(R 3.6.0)
##	ggrepel	0.8.2	2020-03-08	[1]	CRAN	(R 3.6.2)
##	ggsignif	0.6.0	2019-08-08	[1]	CRAN	(R 3.6.0)
##	glue	1.4.0	2020-04-03	[1]	CRAN	(R 3.6.2)
##	gridExtra	* 2.3	2017-09-09	[1]	CRAN	(R 3.6.0)
##	gtable	0.3.0	2019-03-25	[1]	CRAN	(R 3.6.0)
##	gtools	* 3.8.2	2020-03-31	[1]	CRAN	(R 3.6.2)
##	Hmisc	* 4.4-0	2020-03-23	[1]	CRAN	(R 3.6.0)
##	hms	0.5.3	2020-01-08	[1]	CRAN	(R 3.6.0)
##	htmlTable	1.13.3	2019-12-04	[1]	CRAN	(R 3.6.0)
##	htmltools	0.4.0	2019-10-04	[1]	CRAN	(R 3.6.0)
##	htmlwidgets	1.5.1	2019-10-08	[1]	CRAN	(R 3.6.0)
##	httr	1.4.1	2019-08-05	[1]	CRAN	(R 3.6.0)
##	huxtable	* 4.7.1	2020-01-08	[1]	CRAN	(R 3.6.0)
##	igraph	1.2.5	2020-03-19	[1]	CRAN	(R 3.6.0)
##	IRanges	* 2.20.2	2020-01-13	[1]	Bioconductor	
##	iterators	1.0.12	2019-07-26	[1]	CRAN	(R 3.6.0)
##	jpeg	0.1-8.1	2019-10-24	[1]	CRAN	(R 3.6.0)
##	jsonlite	1.6.1	2020-02-02	[1]	CRAN	(R 3.6.0)
##	kableExtra	* 1.1.0	2019-03-16	[1]	CRAN	(R 3.6.0)
##	knitr	* 1.28	2020-02-06	[1]	CRAN	(R 3.6.0)
##	labeling	0.3	2014-08-23	[1]	CRAN	(R 3.6.0)
##	lattice	* 0.20-41	2020-04-02	[1]	CRAN	(R 3.6.2)
##	latticeExtra	0.6-29	2019-12-19	[1]	CRAN	(R 3.6.0)
##	lifecycle	0.2.0	2020-03-06	[1]	CRAN	(R 3.6.0)
##	locfit	1.5-9.4	2020-03-25	[1]	CRAN	(R 3.6.0)
##	magrittr	1.5	2014-11-22	[1]	CRAN	(R 3.6.0)
##	MASS	7.3-51.5	2019-12-20	[1]	CRAN	(R 3.6.0)
##	Matrix	1.2-18	2019-11-27	[1]	CRAN	(R 3.6.2)
##	matrixStats	* 0.56.0	2020-03-13	[1]	CRAN	(R 3.6.0)
##	memoise	1.1.0	2017-04-21	[1]	CRAN	(R 3.6.0)
##	mgcv	1.8-31	2019-11-09	[1]	CRAN	(R 3.6.2)

##	multtest	2.42.0	2019-10-29	[1]	Bioconductor
##	munsell	0.5.0	2018-06-12	[1]	CRAN (R 3.6.0)
##	nlme	3.1-147	2020-04-13	[1]	CRAN (R 3.6.2)
##	nnet	7.3-13	2020-02-25	[1]	CRAN (R 3.6.0)
##	openxlsx	4.1.4	2019-12-06	[1]	CRAN (R 3.6.0)
##	permute	* 0.9-5	2019-03-12	[1]	CRAN (R 3.6.0)
##	phyloseq	* 1.30.0	2019-10-29	[1]	Bioconductor
##	pillar	1.4.3	2019-12-20	[1]	CRAN (R 3.6.0)
##	pkgbuild	1.0.6	2019-10-09	[1]	CRAN (R 3.6.0)
##	pkgconfig	2.0.3	2019-09-22	[1]	CRAN (R 3.6.0)
##	pkgload	1.0.2	2018-10-29	[1]	CRAN (R 3.6.0)
##	plyr	1.8.6	2020-03-03	[1]	CRAN (R 3.6.0)
##	png	0.1-7	2013-12-03	[1]	CRAN (R 3.6.0)
##	prettyunits	1.1.1	2020-01-24	[1]	CRAN (R 3.6.0)
##	processx	3.4.2	2020-02-09	[1]	CRAN (R 3.6.0)
##	ps	1.3.2	2020-02-13	[1]	CRAN (R 3.6.0)
##	purrr	0.3.4	2020-04-17	[1]	CRAN (R 3.6.2)
##	R6	2.4.1	2019-11-12	[1]	CRAN (R 3.6.0)
##	RColorBrewer	1.1-2	2014-12-07	[1]	CRAN (R 3.6.0)
##	Rcpp	1.0.4.6	2020-04-09	[1]	CRAN (R 3.6.2)
##	RCurl	1.98-1.2	2020-04-18	[1]	CRAN (R 3.6.2)
##	readr	1.3.1	2018-12-21	[1]	CRAN (R 3.6.0)
##	remotes	2.1.1	2020-02-15	[1]	CRAN (R 3.6.0)
##	reshape	0.8.8	2018-10-23	[1]	CRAN (R 3.6.0)
##	reshape2	* 1.4.4	2020-04-09	[1]	CRAN (R 3.6.2)
##	rhdf5	2.30.1	2019-11-26	[1]	Bioconductor
##	Rhdf5lib	1.8.0	2019-10-29	[1]	Bioconductor
##	rlang	0.4.5	2020-03-01	[1]	CRAN (R 3.6.0)
##	rmarkdown	2.1	2020-01-20	[1]	CRAN (R 3.6.0)
##	rpart	4.1-15	2019-04-12	[1]	CRAN (R 3.6.2)
##	rprojroot	1.3-2	2018-01-03	[1]	CRAN (R 3.6.0)
##	RSQLite	2.2.0	2020-01-07	[1]	CRAN (R 3.6.0)
##	rstudioapi	0.11	2020-02-07	[1]	CRAN (R 3.6.0)
##	rvest	0.3.5	2019-11-08	[1]	CRAN (R 3.6.0)
##	S4Vectors	* 0.24.4	2020-04-09	[1]	Bioconductor
##	scales	1.1.0	2019-11-18	[1]	CRAN (R 3.6.0)
##	sessioninfo	1.1.1	2018-11-05	[1]	CRAN (R 3.6.0)
##	stringi	1.4.6	2020-02-17	[1]	CRAN (R 3.6.0)
##	stringr	1.4.0	2019-02-10	[1]	CRAN (R 3.6.0)
##	SummarizedExperiment	* 1.16.1	2019-12-19	[1]	Bioconductor
##	survival	* 3.1-12	2020-04-10	[1]	CRAN (R 3.6.2)
##	testthat	2.3.2	2020-03-02	[1]	CRAN (R 3.6.0)
##	tibble	3.0.1	2020-04-20	[1]	CRAN (R 3.6.2)
##	tidyr	1.0.2	2020-01-24	[1]	CRAN (R 3.6.0)
##	tidyselect	1.0.0	2020-01-27	[1]	CRAN (R 3.6.0)
##	usethis	1.6.0	2020-04-09	[1]	CRAN (R 3.6.2)
##	vctrs	0.2.4	2020-03-10	[1]	CRAN (R 3.6.0)
##	vegan	* 2.5-6	2019-09-01	[1]	CRAN (R 3.6.0)
##	viridisLite	0.3.0	2018-02-01	[1]	CRAN (R 3.6.0)
##	webshot	0.5.2	2019-11-22	[1]	CRAN (R 3.6.0)
##	withr	2.2.0	2020-04-20	[1]	CRAN (R 3.6.2)
##	xfun	0.13	2020-04-13	[1]	CRAN (R 3.6.2)
##	XML	3.99-0.3	2020-01-20	[1]	CRAN (R 3.6.0)
##	xml2	1.3.1	2020-04-09	[1]	CRAN (R 3.6.2)

```
## xtable          1.8-4    2019-04-21 [1] CRAN (R 3.6.0)
## XVector         0.26.0   2019-10-29 [1] Bioconductor
## yaml            2.2.1    2020-02-01 [1] CRAN (R 3.6.0)
## zip             2.0.4    2019-09-01 [1] CRAN (R 3.6.0)
## zlibbioc        1.32.0   2019-10-29 [1] Bioconductor
##
## [1] /Library/Frameworks/R.framework/Versions/3.6/Resources/library
```