CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/137377/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Weiser, Rebecca , Rye, Phillip D. and Mahenthiralingam, Eshwar 2021. Implementation of microbiota analysis in clinical trials for cystic fibrosis lung infection: experience from the OligoG phase 2b clinical trials. Journal of Microbiological Methods 181 , 106133. 10.1016/j.mimet.2021.106133

Publishers page: http://dx.doi.org/10.1016/j.mimet.2021.106133

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

Supplementary Data File

Implementation of microbiota analysis in clinical trials for cystic fibrosis lung infection: experience from the OligoG Phase 2b clinical trials

Authors: Rebecca Weiser[1*], Philip D. Rye[2], and Eshwar Mahenthiralingam[1*]

Affiliations:

[1] Microbiomes, Microbes and Informatics Group, Organisms and Environment Division, School of Biosciences, Cardiff University, The Sir Martin Evans Building, Museum Avenue, Cardiff, Wales, CF10 3AX, UK.

[2]AlgiPharma AS, Industriveien 33, N-1337 Sandvika, Norway

*** Corresponding authors:**

Rebecca Weiser, School of Biosciences, Cardiff University, The Sir Martin Evans Building, Museum Avenue, Cardiff, Wales, CF10 3AX, UK.

Co-correspondence: Eshwar Mahenthiralingam, School of Biosciences, Cardiff University, The Sir Martin Evans Building, Museum Avenue, Cardiff, Wales, CF10 3AX, UK.
Tel: +44 (0)29 2087 5875, Fax: +44 (0)29 2087 4305

E-mail addresses: WeiserR@cardiff.ac.uk (R. Weiser), phil.rye@algipharma.com (P.D. Rye) MahenthiralingamE@cardiff.ac.uk (E. Mahenthiralingam)

ORCID ID:

0000-0003-3983-3272 (R. Weiser)
0000-0001-9014-3790 (E. Mahenthiralingam).
0000-0001-7762-3300 (P.D. Rye)

Contents

Supplementary tables

Supplementary Table S1. Analyses conducted on samples in the Bcc trial (13 patients, 155 samples)

Supplementary Table S2. Analyses conducted on samples in the *P. aeruginosa* trial (45 patients, 511 samples)

Supplementary Table S3. Statistical analysis of Shannon diversity at three paired start-end points for the Bcc and *P. aeruginosa* trials

Supplementary Table S4. Statistical analysis of the total abundance key pathogens at three paired start-end points for the Bcc and *P. aeruginosa* trials

Supplementary Table S5. Statistical analysis of the relative abundance of key pathogens at three paired start-end points for the Bcc and *P. aeruginosa* trials

Supplementary Table S6. Supplementary Table 3. The relative and total abundance of *Burkholderia* in Bcc trial samples

Supplementary figures

Supplementary Figure S1. Paired samples from the Bcc and *P. aeruginosa* trials were concordant as shown by high Pearson product-moment correlation coefficients (PPMCC).

Supplementary Figure S2. The lung microbiota in the *P. aeruginosa* trial was linked to the individual rather than treatment.

Supplementary Figure S3. Analysis of microbiota present between paired start and end time-points collected during the Bcc trial.

Supplementary Figure S4. Correlation of loss of bacterial diversity with poor lung function for the Oligo G trial participants.

Supplementary methods

Supplementary method S1: Standardised protocol 1, Sputum sample processing for DNA extraction

Supplementary method S2: Standardised protocol 2, Identification of *Burkholderia* species in sputum samples

Supplementary method S3: Standardised protocol 3, Quantitative PCR (qPCR) using TaqMan probes to determine bacterial load (*P. aeruginosa* and *Burkholderia*)

Supplementary method S4: Standardised protocol 4, Ribosomal RNA Intergenic Spacer Analysis (RISA)

Supplementary method S5: Standardised protocol 5, Bacterial diversity analysis (16S rRNA gene sequencing and analysis)

Supplementary method S6: R scripts for statistical analysis of microbiota data

Supplementary Table S1. Analyses conducted on samples in the *Bcc* trial (13 patients, 155 samples)

Patient	Visit number (paired samples)											
	V1		V2		V4		V5		V7		V8	
	1	2	1	2	1	2	1	2	1	2	1	2
27610-001	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-002	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	-	ABC	A
27610-003	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-004	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-005	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-006	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-007	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-008	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-009	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27610-011	ABCD	A	ABC	A	ABC	A	ABC	A	AB	AC	ABC	A
27611-002	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27611-005	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27611-006	ABCD	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A

Footnotes: **A**, sample subjected to 16s rRNA gene sequencing and used in paired analysis to determine microbiota concordance; **B**, sample used in qPCR analysis; **C**, 16S rRNA sequencing results used to examine genus relative abundance, alpha and beta diversity metrics and for identification of trends in the dataset; **D**, sample used to determine the identity of the infecting *Burkholderia* species by *recA* and *gyrB* gene amplification, sequencing and analysis; '**-**' sample not received and not analysed.

Supplementary Table S2. Analyses conducted on samples in the *P. aeruginosa* trial (45 patients, 511 samples)

Patient	Visit number (paired samples)											
	V1		V2		V4		V5		V7		V8	
	1	2	1	2	1	2	1	2	1	2	1	2
27601-001	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27601-002	ABC	-	ABC	A	ABC	A	ABC	-	ABC	A	ABC	-
27601-004	ABC	A	ABC	-	ABC	A	ABC	A	ABC	-	ABC	A
27602-001	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-003	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-006	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-007	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-008	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-009	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-010	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-011	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27602-013	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27603-001	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27604-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27604-003	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27605-001	ABC	A	ABC	A	ABC	-	ABC	A	ABC	A	ABC	-
27605-002	ABC	-	ABC	A	ABC	A	ABC	A	ABC	-	ABC	A
27606-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27606-003	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27606-004	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27606-005	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27607-001	ABC	A	ABC	A	ABC	-	ABC	A	ABC	A	ABC	A
27607-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
27607-003	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
75201-001	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
75201-004	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
75202-001	ABC	-	ABC	-	ABC	-	ABC	-	ABC	-	ABC	-
75202-002	ABC	-	ABC	-	ABC	-	ABC	-	ABC	-	ABC	-
75202-003	ABC	-	ABC	-	ABC	-	ABC	-	ABC	-	ABC	-
82601-003	ABC	-	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82602-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82602-005	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82602-006	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82602-009	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82602-010	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82603-001	AB	AC	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82603-004	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82604-004	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82604-006	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A

82604-008	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82604-009	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82606-001	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
82608-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A
57801-002	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A	ABC	A

Footnotes: **A**, sample subjected to 16s rRNA gene sequencing and used in paired analysis to determine microbiota concordance; **B**, sample used in qPCR analysis; **C**, 16S rRNA gene sequencing results used to examine genus relative abundance, alpha and beta diversity metrics and for identification of trends in the dataset; '**-**' sample not received and not analysed.

Supplementary Table S3. Statistical analysis of Shannon diversity at three paired start-end points for the Bcc and *P. aeruginosa* trials (Wilcoxon signed-rank test)

Treatment	Test statistic	P-value
Bcc trial		
Start-End	V = 31	P = 0.3396
OligoG Start-End	V = 38	P = 0.6355
Placebo Start-End	V = 42	P = 0. 8394
***P. aeruginosa* trial**		
Start-End	V = 410.5	P = 0.2293
OligoG Start-End	V = 541	P = 0.7971
Placebo Start-End	V = 498	P = 0.8318

Supplementary Table S4. Statistical analysis of the total abundance (qPCR) of key pathogens at three paired start-end points for the Bcc and *P. aeruginosa* trials (Wilcoxon signed-rank test)

Treatment	Test statistic	P-value
Bcc trial – *Burkholderia* total abundance		
Start-End	V = 18	P = 0.05737
OligoG Start-End	V = 51	P = 0.7354
Placebo Start-End	V = 78	P = 0.02148
***P. aeruginosa* trial – *P. aeruginosa* total abundance**		
Start-End	V = 470.5	P = 0.8171
OligoG Start-End	V = 456.5	P = 0.7411
Placebo Start-End	V = 401	P = 0.7071

Footnotes: Statistically significant differences are highlighted in green.

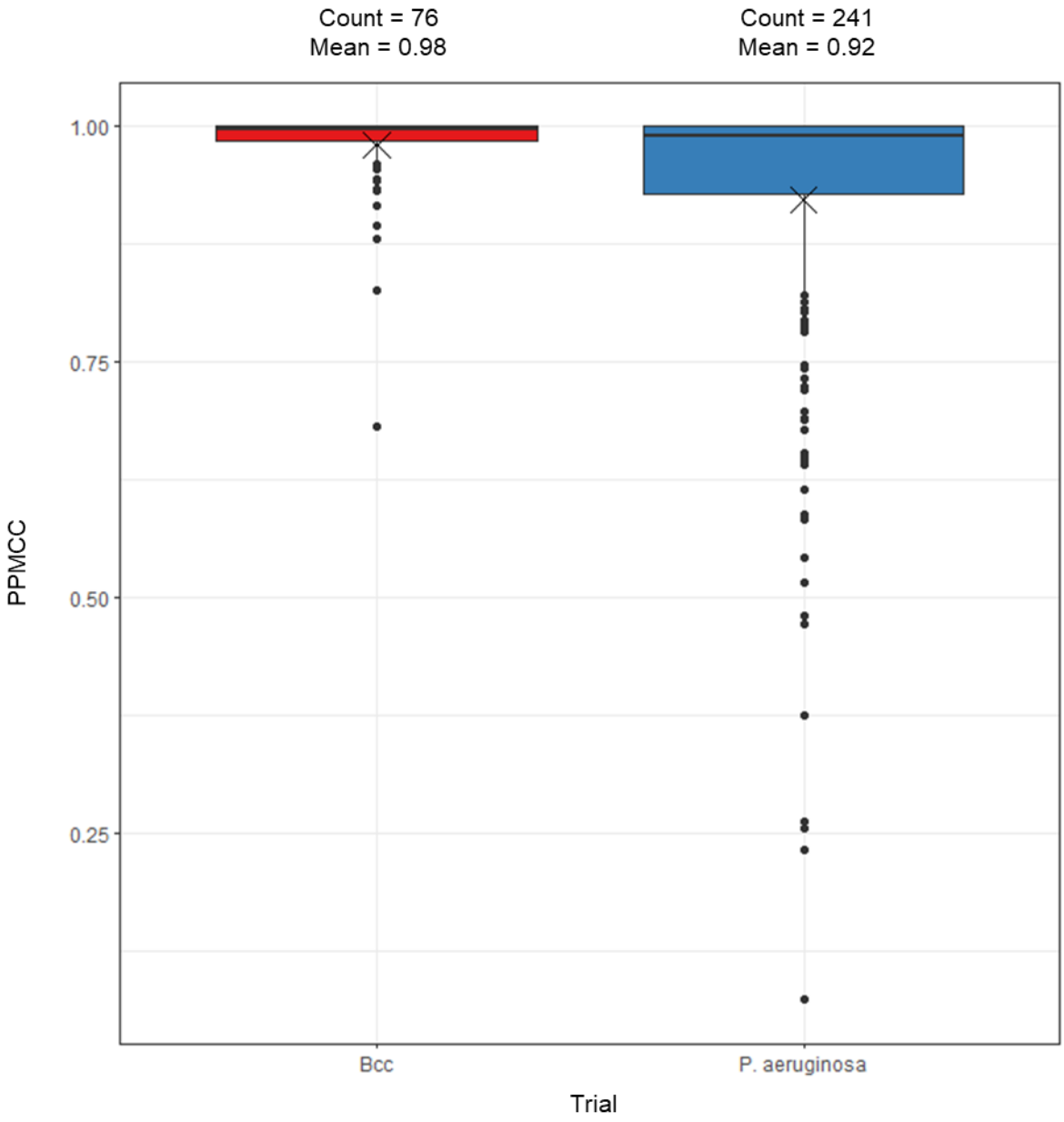
Supplementary Table S5. Statistical analysis of the relative abundance of key pathogens at three paired start-end points for the Bcc and *P. aeruginosa* trials using GAMLSS-BEINF and (mu) logit links

Treatment	Estimate	Std. Error	t value	Pr (>|t|)
Bcc trial – *Burkholderia* relative abundance				
Start-End	0.5107	0.4409	1.158	0.267
OligoG Start-End	0.6461	0.2944	2.195	0.05589
Placebo Start-End	0.5859	0.3010	1.947	0.083261
***P. aeruginosa* trial – *P. aeruginosa* relative abundance**				
Start-End	-0.5372	0.1656	-3.244	0.00226
OligoG Start-End	-0.1030	0.1729	-0.595	0.555
Placebo Start-End	0.2714	0.1757	1.545	0.129

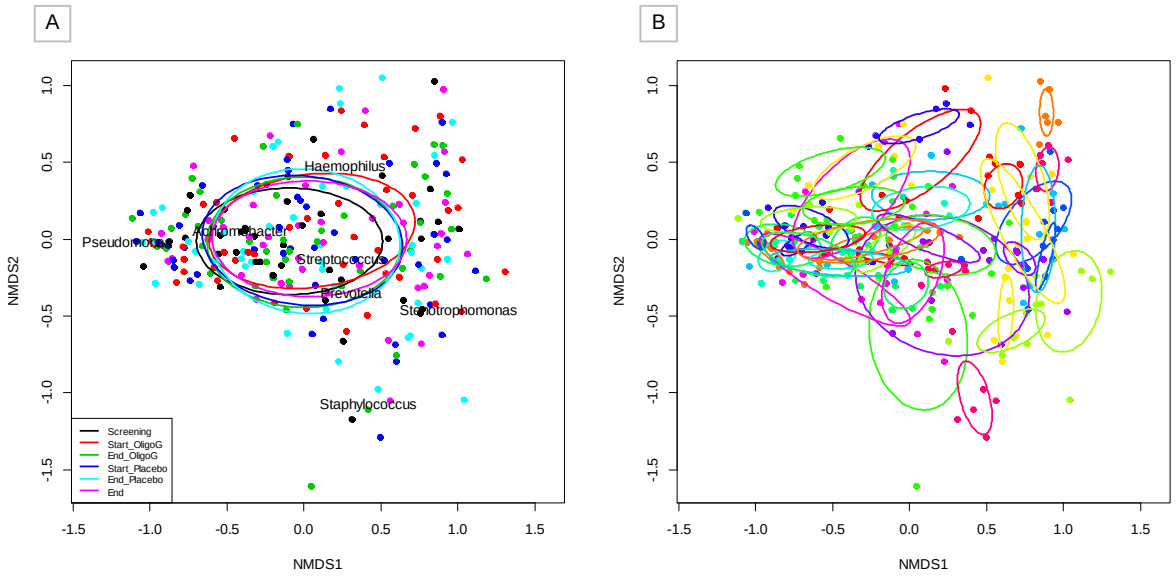
Footnotes: The estimates from the GAMLESS-BEINF models are the difference in log odds of relative abundances between groups. In each model, treatment was the response variable and patient ID was the random effect, with treatment 'start' as the reference class to which treatment 'end' was compared. Statistically significant differences are highlighted in green.

Supplementary Table S6. The relative and total abundance of *Burkholderia* in *Bcc* trial samples

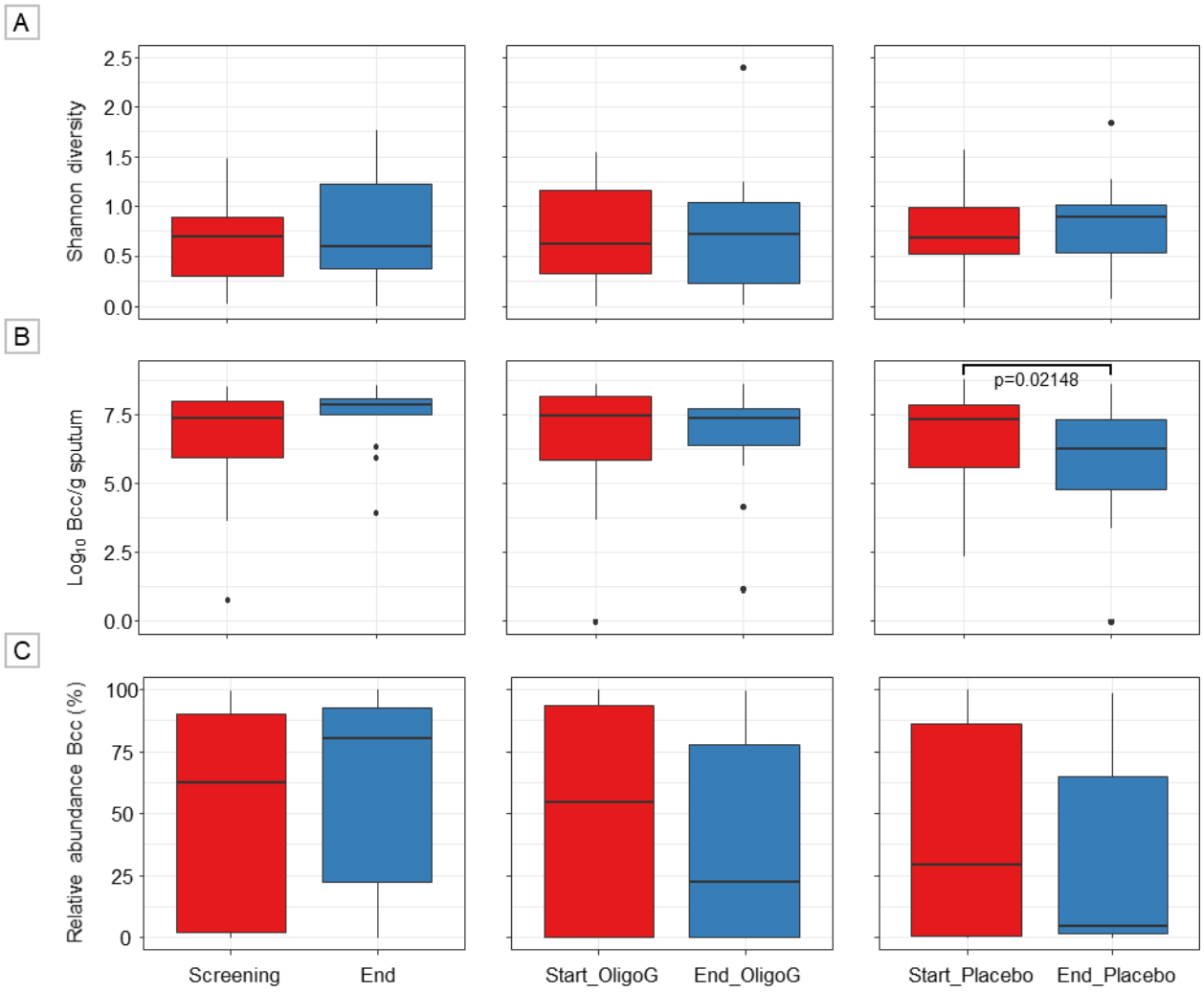
Patient	*Burkholderia* species	Relative abundance (%)	Total abundance (Log Bcc/g sputum)
27610-002	*B. cenocepacia*	90.29	7.74
27610-005		99.62	8.55
27610-006		62.54	8.01
27611-002		66.52	7.36
27610-001	*B. multivorans*	5.2	5.96
27610-004		98.98	8.33
27610-007		8.41	7.50
27610-008		93.56	8.30
27610-009		79.65	6.96
27611-006		2.10	6.41
27610-003	Unknown	0.07	0.76
27610-011		0.00	5.44
27611-005		0.09	3.69



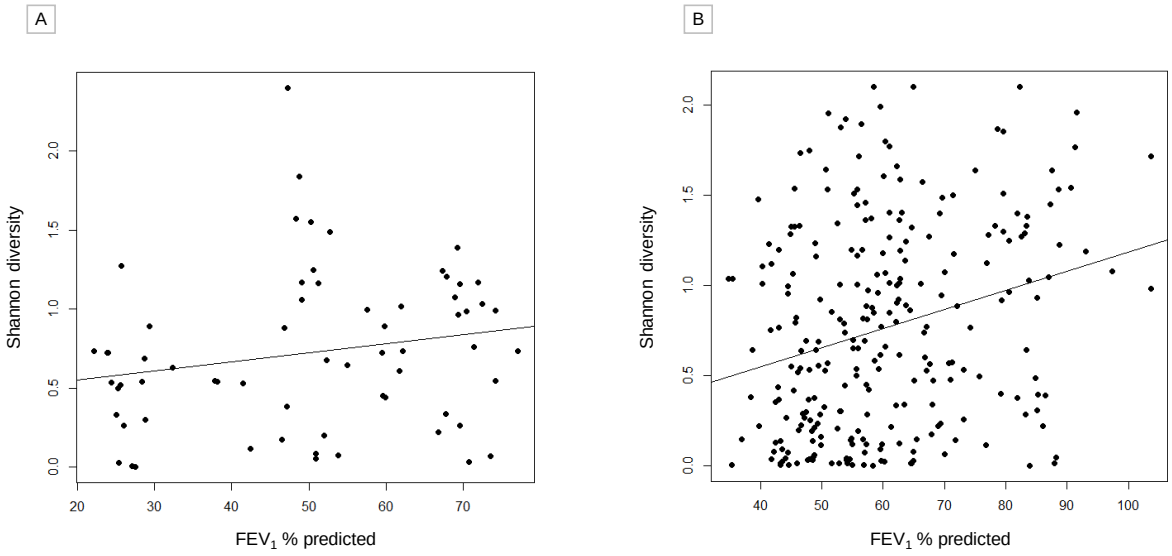
Supplementary Figure S1. Paired samples from the Bcc and *P. aeruginosa* trials were concordant as shown by high Pearson product-moment correlation coefficient (PPMCC) values. The boxplots show the spread of the PPMCC values for each trial, with the mean value highlighted by a cross in the centre of the plots. The number of samples in each trial and the mean values are shown above the plots.



Supplementary Figure S2. The lung microbiota in the *P. aeruginosa* trial was linked to the individual rather than treatment. NMDS analysis of Bray-Curtis dissimilarity values for S1 samples from all 6 time points for the 45 patients on the *P. aeruginosa* trial (except 82601003 that only had S2 for V1) are shown and grouped by treatment **(A)** and patient **(B)**. Points represent individual samples, ellipses are standard deviations of points scores for each grouping. The top 10% genera based on abundance across the whole dataset are shown in **(A)**. A significant difference was observed between patient groups (PERMANOVA, R^2=0.01 p=0.034), but not between treatment groups (PERMANOVA, R^2=0.006, p=0.998). For the treatment sample grouping **(A)** the group variances were homogeneous, satisfying the conditions of the PERMANOVA model. This was not the case for the patient sample grouping **(B)**, indicating that the significant result should be interpreted with caution.



Supplementary Figure S3. Analysis of microbiota present between paired start and end time-points collected during the Bcc trial. Boxplots show the spread of data for Screening versus End samples, Start OligoG versus End OligoG samples, and Start Placebo versus End Placebo (S1 samples only, n=78; except 27610-011 that only had S2 for V7). **(A)** shows the microbiota diversity measured using the Shannon index; **(B)** provides the total abundance of *Burkholderia* per gram of sputum measured using qPCR, and; **(C)** shows the relative abundance of *Burkholderia* from 16S rRNA gene sequencing analysis. For Shannon Diversity and total *Burkholderia* abundance, Wilcoxon signed-rank tests were used to assess the differences between paired time-points. Differences in relative abundance were determined using GAMLESS-BEINF models with patient as the random effect, reporting changes in log(odds ratio) between paired time-points. Statistical significance is shown as a bracket above boxplots with the p-value under the bracket.



Supplementary Figure S4. Correlation of loss of bacterial diversity with poor lung function for the Oligo G trial participants. Linear regression was used to examine the relationship between Shannon diversity and lung function (FEV1 % predicted) for: **(A)** S1 samples (n=78) from all 6 time points for the 13 patients on the Bcc trial (except 27610-011 that only had S2 for V7), and **(B)** S1 samples (n=270) from all 6 time points for the 45 patients on the *P. aeruginosa* trial (except 82601-003 that only had S2 for V1). The regression lines show a trend for decreased lung function with decreased diversity. These trends were not significant when analysed with linear mixed models with patient as the random effect.

Supplementary method S1

Standardised protocol 1: Sputum sample processing for DNA extraction

A) Sample pre-processing

1. Sputum samples must be processed within 4 weeks of sample collection (stored at frozen at -80°C in the meantime)
2. Thaw samples at room temperature for 30-45 minutes

   Label tubes with a Cardiff number and record all sample information

3. Weigh samples using an empty collection tube to tare balance. Record the weight to nearest mg and write on side of tube

   Record sample weights

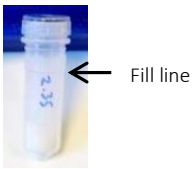
4. Add 4M guanidine isothiocyanate to sample (1:1 ratio, add equivalent ml to weight e.g. if 1 g sample, add 1 ml of guanidine isothiocyante)
5. Centrifuge tubes to bring sputum to the bottom (2 minutes; 1409 *g*)
6. Vortex mix tubes for 30 seconds to 1 minute

If the sputum is still very viscous incubation at 37°C for 30 minutes to 1 hour can be performed to thin the consistency

B) Bead beating

1. Add 1ml of the sputum mix to 1 g of beads in a 2 ml tube with cap and O-ring

   if the sputum is viscous it will be difficult to measure 1 ml accurately, in this case just pipette enough sputum to reach the fill line indicated below



2. Bead beat for 2 minutes on lowest setting of Beadbug (280 x 10 rpm)
3. Pulse centrifuge to settle beads

C) DNA extraction

1. Add 400 µl of sputum mix to a Maxwell16® tissue kit cartridge and run DNA extraction programme on Maxwell16® instrument. In one run, 16 samples can be processed.

   Store remaining sputum mix at -80°C in 1.5 ml non-stick eppendorfs

2. Collect DNA from Maxwell16® instrument into 1.5 ml non-stick eppendorfs

3. Store 20 µl DNA at 4°C for PCR analyses (short term storage)
4. Store remaining DNA at -20°C (long term storage)

D) Reagents/consumables/equipment

Reagent/Consumable/Equipment	Supplier	Product Code
Beadbug microtube homogenizer	Benchmark Scientific	D1030
Triple-pure high impact 100 µm zirconium beads	Benchmark Scientific	D1132-01TP
2 ml bead tubes with caps and seals	Benchmark Scientific	D1031-T20
4 M UltraPure™ Guanidine Isothiocyanate Solution	ThermoFisher Scientific	15577018
Maxwell® 16 System for DNA extraction	Promega	AS1250
Maxwell® Tissue DNA Purification kit (48 preps)	Promega	AS1030

Supplementary method S2

Standardised protocol 2: Identification of *Burkholderia* species in sputum samples

A) PCR amplification of *recA* and *gyrB* gene sequences

Perform separate PCRs for the amplification of *Burkholderia recA* and *gyrB* genes using DNA extracted from sputum samples (Standardised protocol 1).

1. **Order PCR primers** (Spilker *et al.* 2009) from Eurofins Genomics. Primer sequences are shown in Table 1. Prepare stock solutions of individual primers at 100 pmol/µl in nuclease free water according to the synthesis report provided by Eurofins Genomics. Prepare a working solution of a combination of F and R primers at 10 pmol/µl each in nuclease free water (for example 30 µl of F primer and 30 µl R primer in a total volume of 300 µl).

2. **Prepare PCR Mastermix** for the number of reactions required. This will include the number of samples, a positive and negative control, and one additional reaction to account for any pipetting error. Reagent concentrations and volumes for 1 reaction (50 µl) are shown in Table 2.

3. **Run the PCRs** in a thermal cycler with the reaction conditions shown in Table 3. Note that different annealing temperatures are used for *recA* and *gyrB*.

4. **Perform gel electrophoresis** to check for successful amplification of gene products. Prepare a 1.5 % (w/v) gel using molecular grade agarose and Tris-acetate-EDTA (TAE) buffer, stained with SafeView (NBS Biologicals Ltd.; 10 µl SafeView per 100 ml of agarose gel). Load 5-10 µl of PCR product and run in TAE buffer at 80V for approximately 1 hour. Visualise PCR products with a gel imaging system.

B) PCR product purification and sequencing

1. **Purify PCR products** using the QIAquick PCR purification kit (Qiagen Ltd.) according to the manufacturer's instructions. Load the remaining volume from the PCR for each sample (approximately 40 – 45 µl) into the kit.

2. **Quantify PCR products** using the Qubit fluorometer and the Qubit dsDNA BR Assay kit (ThermoFisher Scientific) according to the manufacturer's instructions.

3. **Send the PCR products for sequencing** with the eurofins genomics sequencing service. Note that PCR products may need to be diluted to a certain concentration in nuclease free water before sending, and primers will need to be sent with the PCR products. For both *recA* and *gyrB* send F and R primers to obtain Forward and Reverse sequences for each sample.

C) Analysis of *recA* and *gyrB* sequences

1. **Create consensus *recA* and *gyrB* gene sequences** from the F and R sequence reads. Eurofins genomics will email the gene sequences after sequencing has been completed. Copy and paste the F and R sequences into Notepad and save as .fasta files. Open the .fasta files with the programme BioEdit (Hall 1999).

- Highlight the R sequences and select reverse complement from the Sequence tab (Sequence>Nucleic acid>Reverse Complement)
- Highlight a pair of F and R sequences and create a pairwise alignment (allow ends to slide) of the F and reverse complemented R sequences (Sequence>Pairwise alignment>Align two sequences [allow ends to slide])
- Highlight the F and R sequences and create a consensus sequence (Alignment>Create Consensus Sequence) and save as a .fasta file (File>Export>Split to Individual Fasta Files).

2. **Determine the *Burkholderia* species identity** using the BLAST tool on the *Burkholderia* Genome Database (www.burkholderia.com) (Winsor *et al.* 2008). Search the database using the BLASTN tool and the *recA* and *gyrB* consensus sequences. Include all of the available complete genome sequences in the database in the search. The top database hits will allow the determination of the species identity.

Gene	Gene product	Primers	Sequence 5'>3'	Product size (bp)
recA	Recombinase A	F	AGGACGATTCATGGAAGAWAGC	704
		R	GACGCACYGAYGMRTAGAACTT	
gyrB	DNA gyrase B	F	ACCGGTCTGCAYCACCTCGT	738
		R	YTCGTTGWARCTGTCGTTCCACTGC	

Table 1. PCR primers for the amplification of *recA* and *gyrB* genes

All primers are synthesised and supplied by Eurofins Genomics

Reagent	Final concentration (50 µl)	Volume 1 reaction (µl)
H_20	-	21.6
PCR buffer (10X)	x1	5.0
Q-solution (5X)	x1	10.0
Primers (10 pmol/ul)	1.6 uM	8.0
dNTPs (10 mM each)	240 µm (each)	1.0
Taq	2U	0.4
DNA	-	4

Table 2. PCR reagents for the amplification of *recA* and *gyrB* genes

All reagents are supplied by Qiagen, with the exception of nuclease free H_20 which is supplied by Severn Biotech Ltd.

PCR step	Temperature (°C)	Time (mins)	Cycles
Initial denaturation	95	2	1
Denaturation	94	0.5	30
Annealing	60 (*gyrB*) 58 (*recA*)	0.5	
Extension	72	1	
Final Extension	72	5	1
Indefinite hold	10	-	-

Table 3. PCR reaction conditions for the amplification of *recA* and *gyrB* genes

D) PCR reagents and consumables

Reagent/Consumable	Supplier	Product Code
Taq DNA polymerase (Master Mix kit)	Qiagen	201203
Molecular grade water	Severn Biotech Ltd.	20-9000-01
QIAquick PCR purification kit	Qiagen	28104
Qubit dsDNA BR Assay kit	Invitrogen	Q32853
Qubit Assay tubes	Invitrogen	Q32856

References

Hall, T. A. (1999). BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT. *Nucleic Acids Symposium Series* **41:**95-98.

Spilker, T., Baldwin, A., Bumford, A., Dowson, C. G., Mahenthiralingam, E. and LiPuma, J. J. (2009). Expanded multilocus sequence typing for *Burkholderia* species. *J Clin Microbiol* **47:**2607-2610.

Winsor, G. L., Khaira, B., Van Rossum, T., Lo, R., Whiteside, M. D. and Brinkman, F. S. L. (2008). The *Burkholderia* Genome Database: facilitating flexible queries and comparative analyses. *Bioinformatics* **24:**2803-2804.

Supplementary method S3

Standardised protocol 3: Quantitative PCR (qPCR) using TaqMan probes to determine bacterial load (*Pseudomonas aeruginosa* and *Burkholderia*)

For qPCR background information please refer to the life technologies Real-time PCR handbook, which can be found here: http://www.gene-quantification.com/real-time-pcr-handbook-life-technologies-update-flr.pdf. This protocol uses the Chromo4™ system for real-time detection and the Opticon Monitor Software for analysis. The operation manual can be found here: http://www.bio-rad.com/webroot/web/pdf/lsr/literature/Bulletin_10498.pdf. The *gyrB* gene is used for the quantification of *P. aeruginosa* and the *rpoD* gene is used for *Burkholderia*.

A) qPCR standards

- Reaction efficiency and replicate reproducibility are best assessed through the generation of a standard curve. This is based on a dilution series of the sample nucleic acid which is included in the qPCR run.
- This protocol describes the use of PCR products as standards for qPCR. For this, a DNA fragment larger (>500 bp) than the target qPCR product (generally 100-200 bp) is amplified by PCR, purified and quantified to determine the concentration (and copy number) of the target qPCR product per µl of DNA.

1. Prepare a high concentration 'stock' of the qPCR standard for each gene target

- PCR amplify the fragment of DNA containing the target qPCR gene**.** The PCR reagents, primers and reaction conditions for the amplification of the *rpoD* and *gyrB* genes are shown in Tables 1, 2 and 3. To obtain a large volume of the DNA fragment, run 3 x 50 µl reactions and include a negative control. The species used as positive controls are shown in Table 1.
- Perform standard gel electrophoresis as described in Standardised protocol 2. Load 5 µl of each PCR product.
- Pool the remaining PCR products (45 µl left in each tube, giving a total of approximately 135 µl) and purify using the QIAquick PCR purification kit (Qiagen Ltd.) according to the manufacturer's instructions.
- Quantify PCR products using the Qubit fluorometer (Invitrogen) and Qubit dsDNA BR Assay kit (Invitrogen) according to the manufacturer's instructions to obtain a concentration in ng/µl
- See section E for full details of suppliers of PCR and PCR clean-up and QC reagents
- Calculate the copy number of the target gene using the following equations:

I. (qPCR target bp/DNA fragment bp) x 100 = % of DNA that is qPCR target
II. ng/µl DNA fragment x % of DNA that qPCR target = ng/µl qPCR target
III. [6.023 x 10^{23} (copies/mol) x g/µl qPCR target] /
qPCR target bp x DNA molecular weight = qPCR target copies/µl

Equivalent to:

[6.023 x 10^{14} (Da/ng) x ng/µl qPCR target] /
qPCR target bp x 660 (Da/bp) = qPCR target copies/µl

406

407 ○ A concentration of approximately 10^{10}-10^{11} copies/µl should be obtained from a
408 successful PCR and purification. Store the stock at -20°C.
409
410 **2. Prepare a dilution series**
411 ○ Use nuclease free water to prepare a dilution series from the stock to cover 10^2 – 10^8
412 copies/µl
413 ○ The dilution series should be made up on the day of the qPCR and stored at -20°C
414 for no longer than 24 hours before being discarded. Ideally a fresh dilution series
415 should be made for each qPCR run. Using the Qubit fluorometer, quantify the 10^{10}
416 copies/µl dilution of each dilution series to accurately calculate the copy number for
417 the lower dilutions.

418

419 **Table 1.** PCR primers for the amplification of qPCR standards

Gene	Primers	Sequence 5'>3'	Product size (bp)	Positive control species
rpoD	F	GATCTTGCACATCGTCGTC	1011	*Burkholderia cenocepacia* (J2315)
	R	GTTCGTAACGGAGACGCTG		
gyrB	F	GAGTCGATCACTGTCCGC	1186	*Pseudomonas aeruginosa* (PAO1)
	R	GCATCTTGTCGAAGCGCG		

420 All primers are synthesised and supplied by Eurofins Genomics; refer to Standardised
421 protocol 2 for preparation of primer stock solutions.

422

423 **Table 2.** PCR reagents for the amplification of qPCR standards

424

Reagent	Final concentration	Volume 1 reaction (µl)
H_20	-	21.6
PCR buffer (10X)	x1	5.0
Q-solution (5X)	x1	10.0
Primers (10 pmol/ul)	1.6 uM	8.0
dNTPs (10 mM each)	240 µm (each)	1.0
Taq	2U	0.4
DNA	-	4

425 426 427 428

429 All reagents are supplied by Qiagen, with the exception of nuclease free H_20 which is
430 supplied by Severn Biotech Ltd. Refer to Standardised protocol 2 for preparation of PCR
431 Mastermix.

432

433

Table 3. PCR reaction conditions for the amplification of qPCR standards

PCR step	Temperature (°C)	Time (mins)	Cycles
Initial denaturation	95	2	1
Denaturation	94	0.5	30
Annealing	59 (*rpoD*) 58 (*gyrB*)	0.5	
Extension	72	1	
Final Extension	72	5	1
Indefinite hold	10	-	-

B) qPCR

1. Set up the Chromo4™ real-time detector. Using the Opticon Monitor Software, edit the plate layout and reaction conditions for the qPCR run. An example 96 well-plate layout is shown below in Figure 1. Under Master>Plate Setup>Edit>Specify Quant Standards, the copies/µl of each of the standards can be entered. Under Master>Plate Setup>Edit>Sets, replicates can be grouped together into sets and given an ID. The reaction conditions for the 16S rRNA, *rpoD* and *gyrB* gene qPCRs are shown in Table 4.

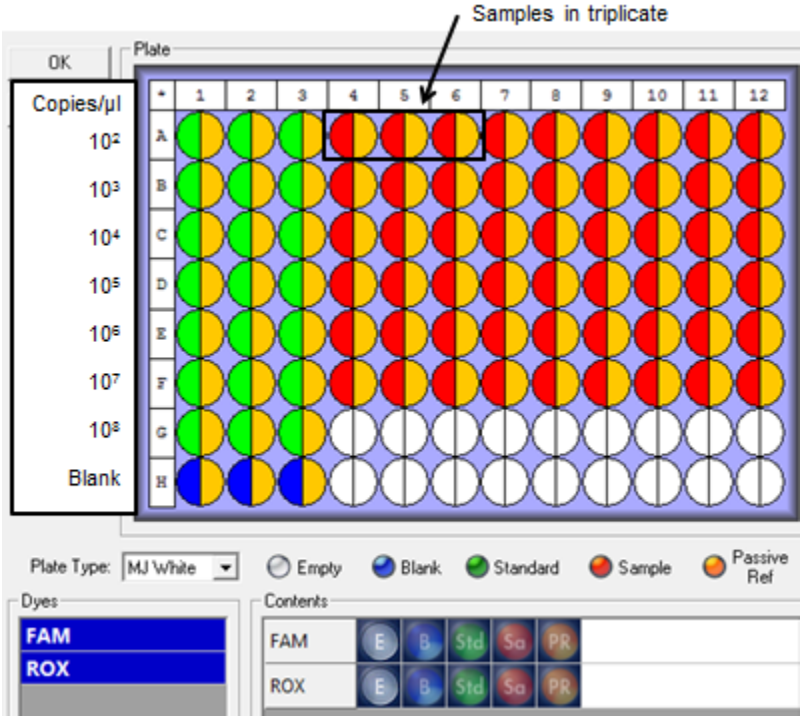


Figure 1. Example qPCR plate layout on the Opticon Monitor Software. Standards, samples and blanks are run in triplicate. The passive reference (PR) dye ROX is a component of the qPCR Mastermix and this is highlighted for each well.

Table 4. qPCR reaction conditions for the amplification of *rpoD* and *gyrB* genes

Step	Time	Temp (°C)	Nº cycles
UNG treatment	3 min	50	1
Taq activation	10 min	95	1
Denaturation	15 secs	95	40
Annealing and Extension	30 secs (*rpoD*) 30 secs (*gyrB*)	67 60	
Plate Read			
Hold	10 mins	25	1

2. Prepare the qPCR Mastermix for the number of reactions required, bearing in mind that everything is performed in triplicate. This will include the 10^2 – 10^8 dilution series, the samples and the blanks. Add an extra reaction to account for any pipetting error. The qPCR primers and TaqMan probes, and reagents are shown in Tables 5 and 6, respectively.
3. Load the Mastermix into the qPCR plate, 9 µl into each well.
4. Load the standards, blanks and samples into the qPCR plate, 1 µl into each well.
5. Seal the plate with a plate seal and pulse centrifuge the plate to draw the liquid to the bottom of the wells
6. Transfer the plate into the qPCR machine and run the qPCR. The qPCR plate can be discarded once the run is complete.

Table 5. qPCR primers and TaqMan probes for the amplification of *rpoD* and *gyrB* genes

Gene	Primers/Probe	Sequence 5'>3'	Product size (bp)	Reference
rpoD	F	GAGATGAGCACCGATCACAC	143	(Sass *et al.* 2013)
	R	CCTTCGAGGAACGACTTCAG		
	PROBE	5'FAM-CTGCGCAAGCTGCGTCACC-3'MGBNFQ		This study
gyrB	F	CCTGACCATCCGTCGCCACAAC	220	(Anuj *et al.* 2009)
	R	CGCAGCAGGATGCCGACGCC		
	PROBE	5'FAM-GGTCTGGGAACAGGTCTACCACCACGG-3'MGBNFQ		

Table 6. qPCR Mastermix: reagents for the amplification of 16S rRNA, *rpoD* and *gyrB* genes

Reagent	Final concentration	Volume 1 reaction (µl)
qPCR Supermix with ROX (2X)*	1X	5
F&R primers (10 pmol/µl)	1.8 µM	1.8
TaqMan Probe (100 pmol/µl) [FAM]*	225 nM	0.0225
H_2O	-	2.1775

*See Section E for the suppliers of the qPCR Supermix and TaqMan Probes

C. Analysis of qPCR results in Opticon Monitor Software

- In the Opticon Monitor Software click the 'Quantitation' view to see the results. This view will show you the plate layout, wells and sets, and the data and standards graphs. An example is shown in Figure 2.
- To calculate the efficiency of the qPCR amplification, use the slope of the standard curve and the below equation. The 'Toggle Axis' box needs to be checked to get the correct x value to use in the equation. The acceptable range is 90 -110%.

qPCR reaction efficiency = $[10^{(-1/\text{slope of standard curve})} - 1] \times 100$

qPCR reaction efficiency (Figure 2) = $[10^{(-1/-3.429)} - 1] \times 100 = 95.7\%$

- The R^2 value is a measure of replicate reproducibility and should ideally be above 0.98 (98%). The R^2 value from Figure 2 is 0.999 (99.9%)
- If you are satisfied with the efficiency and R^2 values move on to processing the data

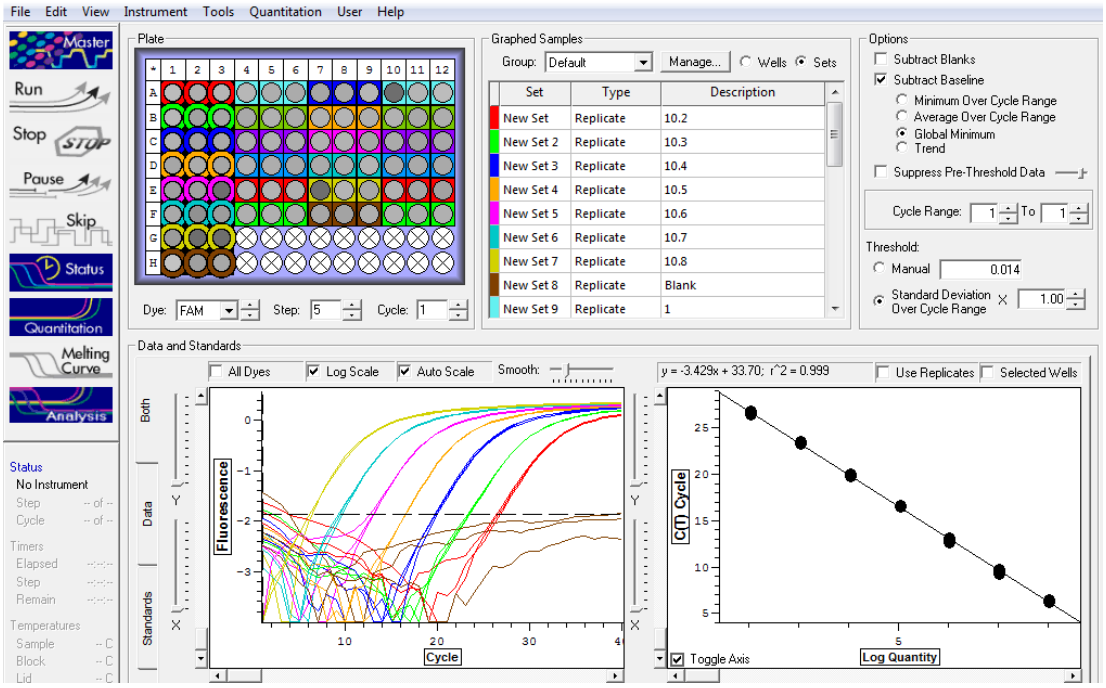


Figure 2. qPCR results under the 'Quantitation' view in Opticon Monitor Software. The data graph in the bottom left hand window displays the fluorescence curves of the standards, the bottom right hand window displays the standard curve from which the efficiency and R^2 values can be obtained.

- To view the results from all of the wells, highlight all of the wells in the 'Plate' section, and all of the sets in the 'Graphed Samples' section. Select Quantitation>Copy to Clipboard>Quantity calculations, and paste this into an Excel spreadsheet.
- Look at the triplicate results for each sample and calculate the coefficient of variation (CV = standard deviation/mean). If the CV is equal to or less than 20% then the mean of the three results can be taken. If the CV is higher than 20% examine the results

further. If two results within a triplicate set are within 0.2 log of each other, the mean of these two results is taken, and the third 'anomalous' result is excluded from the dataset. If assays do not meet these criteria, they are considered unacceptable and the qPCR re-run for the sample. A full explanation of these criteria can be found in (Zemanick *et al.* 2010).

- Opticon Monitor Software will perform preliminary analysis of replicate sets for you, allowing you to observe the average C(t), the average copy number, and some basic statistics for each set of replicates. Remove wells with anomalous results from the sets under the Master>Plate Setup>Edit>Sets, return to the Quantitation view, highlight all of the sets in the Graphed Samples section, and select Quantitation>Copy to Clipboard>Quantity calculations, and paste this into an Excel spreadsheet.

D. Statistical analysis of qPCR results

- There should be at least 3 qPCR runs (3 biological replicates) per sample. These results can be used for further statistical analyses.
- The results obtained are gene copy per µl of DNA (equivalent to cell number) extracted from a sputum sample. Perform the necessary calculations to obtain the number of gene copies (cell number) per gram of sputum for comparisons.

E. Reagents and consumables

Table 7. qPCR Reagents and consumables

Reagent/Consumable	Supplier	Product Code
Preparation of qPCR standards		
Taq DNA polymerase (Master Mix kit)	Qiagen	201203
Molecular grade water	Severn Biotech Ltd.	20-9000-01
QIAquick PCR purification kit	Qiagen	28104
Qubit dsDNA BR Assay kit	Invitrogen	Q32853
Qubit Assay tubes	Invitrogen	Q32856
qPCR		
Platinum qPCR supermix-UDG w/Rox	Life Technologies (Thermo Fisher Scientific)	11743-500
Custom TaqMan probe (5'FAM, 3'MGBNFQ)	ThermoFisher Scientific	Order online: https://www.thermofisher.com/order/custom-oligo/custom-taqman-probes
Hard-Shell low-profile Thin-wall 96-well skirted PCR plates (black shell white well)	BioRad	HSP9665
Sealing film for real time PCR (50 µm thick)	ELKAY	SEA-LPTS-RT2

References

Anuj, S. N., Whiley, D. M., Kidd, T. J., Bell, S. C., Wainwright, C. E., Nissen, M. D. and Sloots, T. P. (2009). Identification of *Pseudomonas aeruginosa* by a duplex real-time polymerase chain reaction assay targeting the ecfX and the gyrB genes. *Diagnostic Microbiology and Infectious Disease* **63:**127-131.

Sass, A. M., Schmerk, C., Agnoli, K., Norville, P. J., Eberl, L., Valvano, M. A. and Mahenthiralingam, E. (2013). The unexpected discovery of a novel low-oxygen-activated locus for the anoxic persistence of Burkholderia cenocepacia. *Isme j* **7:**1568-1581.

Zemanick, E. T., Wagner, B. D., Sagel, S. D., Stevens, M. J., Accurso, F. J. and Harris, J. K. (2010). Reliability of quantitative real-time PCR for bacterial detection in cystic fibrosis airway specimens. *PLoS One* **5**.

Supplementary method S4

Standardised protocol 4: Ribosomal RNA Intergenic Spacer Analysis (RISA)

The region between the small and large subunit rRNA genes, the 16S rRNA and 23S rRNA gene for bacteria, is known as the Intergenic Transcribed Spacer (ITS). This region varies in length and sequence and is the basis of RISA (Fisher and Triplett 1999). RISA profiles provide the following information about bacterial diversity in polymicrobial samples: (i) the number of RISA bands is a qualitative measure of the taxonomic diversity present; (ii) the size of the ITS amplicon may be correlated to a known species/genera, especially if run alongside a positive control, and provide a presumptive taxonomic identification; and (iii) the RISA profiles are semi-quantitative, with the ITS band intensity correlating to the amount of DNA template and hence the approximate proportion of the original organism present in the sputum sample.

This protocol describes the use of RISA to check quality of the DNA extracted from sputum (Standardised protocol 1), i.e. to determine if there is sufficient bacterial DNA for a positive PCR result, and to get a preliminary idea of the bacterial diversity in a sputum sample.

A) RISA-PCR

5. RISA-PCR primers are shown in Table 1, please refer to Standardised protocol 2 for more details on primer stock preparation.
6. Prepare the RISA-PCR Mastermix for the number of reactions required. This will include the number of samples, a positive and negative control, and one additional reaction to account for any pipetting error. PCR primers, reagent concentrations and volumes for 1 reaction (25 µl) are shown in Table 2. Control DNA from *Pseudomonas aeruginosa*, *Burkholderia spp.* and other bacterial species linked with cystic fibrosis can also be run to produce control products for profile analysis.
7. Run the RISA-PCR in a thermal cycler with the PCR conditions shown in Table 3.
8. Perform standard gel electrophoresis to check for successful amplification of PCR products

Table 1. PCR primers for the amplification of the bacterial ITS region

Primer	Sequence (5' > 3')	Comment	Reference
1406F	TGYACACACCGCCCGT	Degenerate RISA primers	Fisher & Triplett (1999)
23SR	GGGTTBCCCCATTCRG		

All primers are synthesised and supplied by Eurofins Genomics

Table 2. PCR reagents for the amplification of the bacterial ITS region

Reagent	Final concentration (25 ul)	Volume 1 reaction (ul)
H_20	-	13.8
PCR buffer (10X)	x1	2.5
Q-solution (5X)	x1	5.0
Primers (10 pmol/ul)	0.4 uM	1.0
dNTPs (10 mM each)	240 um (each)	0.5
Taq	2U	0.2
DNA	-	2

All reagents are supplied by Qiagen, with the exception of nuclease free H_20 which is supplied by Severn Biotech Ltd.

Table 3. PCR reaction conditions for the amplification of the bacterial ITS region

PCR step	Temperature (°C)	Time (mins)	Cycles
Initial denaturation	95	5	1
Denaturation	95	1	35
Annealing	54	0.5	
Extension	72	1	
Final Extension	72	5	1
Indefinite hold	10	-	-

B) Microfluidic separation and cluster analysis of RISA-PCR profiles

1. Run RISA-PCR products on an Agilent BioAnalyzer (Agilent Technologies UK Ltd., Cheshire, United Kingdom) using a DNA 7500 chip according to the manufacturer's instructions.
2. Import BioAnalyzer profiles into Bionumerics software (Applied Maths, Gent, Belgium) for analysis. A dedicated script has been provided to Cardiff University by Applied Maths to convert BioAnalyzer profiles to a format compatible with Bionumerics.
3. Cluster analysis is performed in Bionumerics to compare the RISA-PCR profiles. Similarities between RISA are calculated using the Pearson coefficient, and dendrograms constructed by the unweighted-pair group method using average linkages (UPGMA). It is also possible to view the profiles as a composite image to look at the patient samples in chronological order. Changes in overall diversity can be observed by looking at the number and sizes of bands within the profile.
4. The size of specific RISA amplicons can be determined using the BioAnalyzer and correlated to specific bacterial genera using the In Silico PCR database (website http://insilico.ehu.es/PCR)(Bikandi *et al.* 2004). The intergenic spacer sizes can be recorded for strains and species of interest (examples are shown in Table 4). It is not

always possible to identify bands within the RISA profile as certain species, but bands of certain sizes can be useful markers to look out for.

Table 4. Intergenic transcribed spacer (ITS) sizes (bp) for a selection of bacterial species and strains associated with cystic fibrosis

Species/strain	ITS size(s) (bp)
Pseudomonas aeruginosa	753
Burkholderia ambifaria AMMD	681, 827, 830, 871
Burkholderia cenocepacia J2315	677, 812, 816
Burkholderia multivorans ATCC 17616	829, 830, 883
Haemophilus influenzae	995-1015, 757-759
Ralstonia mannitolytica	805
Ralstonia pickettii	796-797
Stenotrophomonas maltophilia	777-782, 826-829
Achromobacter xylosoxidans	887-891, 1021

C) Reagents/consumables/equipment

Reagent/Consumable	Supplier	Product Code
Taq DNA polymerase (Master Mix kit)	Qiagen	201203
Molecular grade water	Severn Biotech Ltd.	20-9000-01
2100 Bioanalyzer instrument	Agilent	G2939B
Agilent DNA 7500 kit	Agilent	5067-1506
Agilent 7500 reagents	Agilent	5067-1507

References

Bikandi, J., San Millán, R., Rementeria, A. and Garaizar, J. (2004). In silico analysis of complete bacterial genomes: PCR, AFLP–PCR and endonuclease restriction. *Bioinformatics* **20:**798-799.

Fisher, M. M. and Triplett, E. W. (1999). Automated approach for ribosomal intergenic spacer analysis of microbial diversity and its application to freshwater bacterial communities. *Applied and Environmental Microbiology* **65:**4630-4636.

\

Supplementary method S5

Standardised protocol 5: Bacterial diversity analysis (16S rRNA gene sequencing and analysis)

A) **Extract DNA** from the samples to be sent for 16S rRNA sequencing and bacterial diversity analysis (Standardised protocol 1).

B) **Check DNA quality** by performing a RISA PCR (Standardised protocol 3). This will determine whether there is sufficient DNA for a positive PCR result and will also give some preliminary information about bacterial diversity.

C) **Sample submission** to the commercial company Research and Testing Laboratory Inc. (Lubbock, Texas, USA; http://www.researchandtesting.com). Sample submission guidelines can be found online and a submission spreadsheet will need to be completed and sent back. Samples are sent via a courier to the company. Research and Testing perform Illumina MiSeq sequencing of the 16S rRNA gene V1-V2 regions and on completion will email a web link which can be used to retrieve the data.

D) **Data processing and analysis in Mothur (version 1.33)**

Mothur is a bioinformatics program for analysing microbial communities that is being developed by Schloss lab at the University of Michigan. This protocol has been adapted from the MiSeq SOP found at http://www.mothur.org/wiki/MiSeq_SOP#OTU-based_analysis to process 16S rRNA gene sequences that are generated using Illimina's MiSeq platform using paired end reads. Please refer to this website for a more detailed description of the pipeline steps.

1. Preparing the files for Mothur

Based on Illumina's MiSeq platform, there are two fastq files generated for each sample. The R1 file corresponds to read 1 (forward read) and the R2 file corresponds to read 2 (reverse read). It may be useful to rename these files (e.g. example_R1, example_R2) before starting the Mothur pipeline as the original file names can be very long.

These fastq files then need to be paired up for each sample. Using Microsoft Excel in windows, make a spreadsheet with 3 columns but no headings. Column A is sample name, column B is R1 file name and column C is R2 file name. Save this as a .txt. file. This file specifies which fastq files to pair together. Copy the .txt file and all of the sequence files over to the directory that you will be using on the linux platform.

- **Move to the linux platform** and navigate to the directory where your sequence files and the .txt file that you have just created is stored. In Mothur use the **make.contigs** command to extract the sequence and quality score data from your fastq files, create the reverse complement of the reverse read and then join the reads into contigs.

```
mothur > make.contigs (file=example.txt, processors=4)
```

This command also produces several files that you will need later: example.trim.contigs.fasta and example.contigs.groups. These contain the sequence data and group identity for each sequence. Note that 'groups' actually refers to samples in Mothur. The example.contigs.report file will tell you something about the contig assembly for each read.

Look at a summary of the sequences:

```
mothur > summary.seqs (fasta=example.trim.contigs.fasta, processors=4)
```

Example output:

```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       35      35      0       3       1
2.5%-tile:      1       342     342     0       5       214983
25%-tile:       1       343     343     0       5       2149830
Median:         1       344     344     0       5       4299659
75%-tile:       1       348     348     0       5       6449488
97.5%-tile:     1       367     367     8       5       8384334
Maximum:        1       502     502     331     249     8599316
Mean:   1       347.077 347.077 0.881615        5.26081
# of Seqs:      8599316
```

2. Reducing sequencing and PCR errors

- Remove sequences with ambiguous bases and filter sequences based on different criteria using the **screen.seqs** command. The criteria will vary depending on the sequence data.

```
mothur > screen.seqs(fasta=example.trim.contigs.fasta, group=example.contigs.groups,
summary=example.trim.contigs.summary, maxn=0, maxambig=0, maxhomop=7,
minlength=342, maxlength=360)
```

- Re-summarise the sequences after filtering

```
mothur > summary.seqs (fasta=example.trim.contigs.good.fasta, processors=4)
```

Example output:

```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       342     342     0       3       1
2.5%-tile:      1       342     342     0       5       173065
25%-tile:       1       344     344     0       5       1730650
Median:         1       344     344     0       5       3461300
75%-tile:       1       344     344     0       5       5191950
97.5%-tile:     1       355     355     0       5       6749535
Maximum:        1       360     360     0       7       6922599
Mean:   1       345.554 345.554 0       5.00096
# of Seqs:      6922599
```

3. Processing improved sequences

- Remove duplicate sequences using the **unique.seqs** command.

```
mothur > unique.seqs(fasta=example.trim.contigs.good.fasta)
```

- Run the **count.seqs** command to generate a table where the rows are the names of the unique sequences and the columns are the names of the groups. The table is then filled with the number of times each unique sequence shows up in each group.

```
mothur > count.seqs(name=example.trim.contigs.good.names,
group=example.contigs.good.groups)
```

- Re-summarise the sequences and observe the number of unique sequences

```
mothur > summary.seqs (count=example.trim.contigs.good.count_table)
```

Example output:

```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       342     342     0       3       1
2.5%-tile:      1       342     342     0       5       173065
25%-tile:       1       344     344     0       5       1730650
Median:         1       344     344     0       5       3461300
75%-tile:       1       344     344     0       5       5191950
97.5%-tile:     1       355     355     0       5       6749535
Maximum:        1       360     360     0       7       6922599
Mean:   1       345.554 345.554 0       5.00096
# of unique seqs:       840523
total # of seqs:        6922599
```

- View the number of sequences in each sample. The following command will output a list of samples and the number of sequence reads per sample.

```
mothur > count.groups(count=example.trim.contigs.good.count_table)
```

- Align the sequences to a customised reference alignment of bacterial sequences. This reference alignment (silva.bacteria.fasta) must be downloaded into the directory you are working from and is available here: http://www.mothur.org/wiki/Silva_reference_files

```
mothur > align.seqs(fasta=example.trim.contigs.good.unique.fasta,
reference=silva.bacteria.fasta, processors=10
```

- Summarise the quality of the sequences after making the alignment

```
mothur > summary.seqs(fasta=example.trim.contigs.good.unique.align,
count=example.trim.contigs.good.count_table, processors=10)
```

```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1044    1062    1       0       1       1
2.5%-tile:      1044    6424    323     0       5       173065
25%-tile:       1044    6424    325     0       5       1730650
Median:         1044    6424    325     0       5       3461300
75%-tile:       1044    6424    325     0       5       5191950
97.5%-tile:     1044    6424    336     0       5       6749535
Maximum:        43116   43116   359     0       7       6922599
Mean:   1044.1  6424.06 326.565 0       5.00094
# of unique seqs:       840523
total # of seqs:        6922599
```

- Some sequences may start and end at the same position indicating a very poor alignment, which is generally due to non-specific amplification. To make sure that everything overlaps the same region re-run **screen.seqs** to get rid of sequences that start at or before position 1044 and end at or after position 6424, i.e. the start and end values that correspond to the 25%-tile and 75%-tile.
- Include the count table to update the table for the sequences we are removing. Include the summary file so we do not have to figure out again all the start and stop positions.

```
mothur > screen.seqs(fasta=example.trim.contigs.good.unique.align,
count=example.trim.contigs.good.count_table,
summary=example.trim.contigs.good.unique.summary, start=1044, end=6424,
maxhomop=7)
```

- Summarise the sequence data

```
mothur > summary.seqs(fasta=current, count=current)
```

Example output:

```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1044    6424    293     0       3       1
2.5%-tile:      1044    6424    323     0       5       172521
25%-tile:       1044    6424    325     0       5       1725204
Median:         1044    6424    325     0       5       3450407
75%-tile:       1044    6424    325     0       5       5175610
97.5%-tile:     1044    6424    336     0       5       6728292
Maximum:        1044    7694    355     0       7       6900812
Mean:   1044    6424    326.565 0       5.0008
# of unique seqs:      827648
total # of seqs:        6900812
```

- Filter the sequences to remove the overhangs at both ends even though paired-end sequencing should not be an issue. In addition, any column that contains gap characters e.g. “-“will be removed
- The command **filter.seqs** removes columns from alignments based on a criteria defined by the user. For example, alignments generated against reference alignments (e.g. from RDP, SILVA, or greengenes) often have columns where every character is either a '.' or a '-'. These columns are not included in calculating distances because they have no information in them.
- Vertical = T: any column that only contains gap characters is ignored. This can be turned off by setting vertical to F.
- Trump =.: trump option will remove a column if the trump character is found at that position in any sequence of the alignment.

```
mothur > filter.seqs(fasta=example.trim.contigs.good.unique.good.align, vertical=T,
trump=.)
```

Example output:

```
Length of filtered alignment: 804
Number of columns removed: 49196
Length of the original alignment: 50000
Number of sequences used to construct filter: 827648
```

- According to the output, this means that our initial alignment was 50000 columns wide and that we were able to remove 49196 terminal gap characters using trump=. and vertical gap characters using vertical=T. The final alignment length is 804 columns. Because some redundancy has been created across our sequences by trimming the ends, re-run **unique.seqs**

```
mothur > unique.seqs(fasta=example.trim.contigs.good.unique.good.filter.fasta,
count=example.trim.contigs.good.good.count_table)
```

- Further de-noise the sequences using the **pre.cluster** command, which implements a pseudo-single linkage algorithm with the goal of removing sequences that are likely due to sequencing errors

```
mothur > pre.cluster(fasta=example.trim.contigs.good.unique.good.filter.unique.fasta,
count=example.trim.contigs.good.unique.good.filter.count_table, diffs=2, processors=10)

mothur >
chimera.uchime(fasta=example.trim.contigs.good.unique.good.filter.unique.precluster.fasta,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.count_table,
dereplicate=t)

mothur >
remove.seqs(fasta=example.trim.contigs.good.unique.good.filter.unique.precluster.fasta,
accnos=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.accnos)
```

Example output:

```
Removed 43166 sequences from your fasta file.
```

- Re-summarise the sequence data

```
                Start   End     NBases  Ambigs  Polymer NumSeqs
Minimum:        1       804     293     0       3       1
2.5%-tile:      1       804     323     0       5       163245
25%-tile:       1       804     325     0       5       1632445
Median:         1       804     325     0       5       3264889
75%-tile:       1       804     325     0       5       4897333
97.5%-tile:     1       804     336     0       5       6366533
Maximum:        1       804     355     0       7       6529777
Mean:   1       804     326.424 0       4.99811
# of unique seqs:       148203
total # of seqs:        6529777
```

- The **split.abund** command reads a fasta file and a list or a names file splits the sequences into rare and abundant groups.

```
mothur >
split.abund(fasta=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.fasta,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.count_tab
le, cutoff=2)
```

- The **classify.seqs** command allows you to use several different methods to assign their sequences to the taxonomy outline of their choice. Current methods include using a k-

nearest neighbor consensus and Wang approach. The command requires that you provide a fasta-formatted input and database sequence file (trainset10_082014.rdp.fasta) and a taxonomy file (trainset10_082014.rdp.tax) for the reference sequences. These can be downloaded from the following website: http://www.mothur.org/wiki/RDP_reference_files

```
mothur >
classify.seqs(fasta=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.fasta,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.abund.count_table, reference=trainset10_082014.rdp.fasta, taxonomy=trainset10_082014.rdp.tax,
cutoff=80, processors=10)
```

- Next, since bacterial 16S rRNA sequences were amplified, anything classified as species other than bacteria (e.g. mitochondria, archaea, etc.) have to be removed.
- The **remove.lineage** command reads a taxonomy file and a taxon and generates a new file that contains only the sequences not containing that taxon. You may also include either a fasta, name, group, list, count or align.report file to this command and Mothur will generate new files for each of those that contains only the sequences not containing that taxon.

```
mothur >
remove.lineage(fasta=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.fasta,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.abund.count_table,
taxonomy=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.rdp.wang.taxonomy, taxon=Chloroplast-Mitochondria-unknown-Archaea-Eukaryota)
```

- The **cluster.split** command can be used to assign sequences to OTUs and outputs a .list, .rabund, .sabund files. It splits large distance matrices into smaller pieces

```
mothur >
cluster.split(fasta=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.fasta,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.abund.pick.count_table,
taxonomy=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.rdp.wang.pick.taxonomy, splitmethod=classify, taxlevel=4, cutoff=0.15, processors=10)
```

- Observe the number of sequences left

```
mothur > count.groups(count=current)
```

4. Preparing for analysis

- Create a shared file, which has samples in columns and OTUs in rows and displays the count of each OTU in each sample. From this we can see how many sequences are in each OTU from each group and tell Mothur that we're really only interested in the 0.03 cutoff level:

```
mothur >
make.shared(list=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.an.unique_list.list,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.abund.pick.count_table, label=0.03)
```

- Get the consensus taxonomy for each OTU using the **classify.otu** command

```
mothur >
classify.otu(list=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.an.unique_list.list,
count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.abund.pick.count_table,
taxonomy=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.rdp.wang.pick.taxonomy, label=0.03)
```

- Check the number of sequences in each sample

```
mothur >
count.groups(count=example.trim.contigs.good.unique.good.filter.unique.precluster.uchime.pick.abund.pick.count_table)
```

- Subsampling the data ensures that each sample has the same number of reads for downstream analyses. This is done by **sub.sample**. At this point, you need to decide what to normalize all the reads in each sample to. If you do not specify "size=…" in the command, then Mothur will look at all your groups and find the one with the lowest number of reads and sub-sample everything to it, this is shown below:

```
mothur >
sub.sample(shared=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.an.unique_list.shared)
```

- In the case that there is a sample with an extremely low number of read, specify and subsample by a suitable size by changing XXX in the following command to the size that you prefer in this command (the minimum read number used should be >1000):

```
mothur >
sub.sample(shared=example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.an.unique_list.shared, size=XXX
```

- Rename the very long file names to simpler ones. This can be done by the following command, which basically makes a copy of the file using the name you specified. The final files will all have .pick.pick near the end. Do this for the .groups, .list, and subample.shared files. The final .fasta file will come from a subsequent step (the get.oturep step in stage 5). The example below shows the command for renaming the .list file:

```
mothur > system (rename
example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.an.unique_list.list example.final.list)
```

- Alternatively, quit mothur and use the cp command to copy of the file and rename the copy you have made. You will need to start the mothur software again once you have copied the files. Example commands are shown below. The first command exits mothur, the second command copies and renames the file (.list in this example), and the third command re-starts mothur.

```
mothur > quit
> cp
example.trim.contigs.good.unique.good.filter.unique.precluster.pick.abund.pick.an.unique_list.list example.final.list
> mothur
```

5. Analysis

- Obtain a summary file. The **summary.single** command produces a table containing the number of sequences, alpha diversity indices and the sample coverage. Although the downstream analyses (alpha and beta diversity) will be performed using R statistical software (R-Core-Team 2013)(http://www.R-project.org), this summary is useful to gain an overview of the sample statistics.

```
mothur > summary.single(shared=example.final.shared, calc=nseqs-sobs-chao-ace-invsimpson-npshannon-coverage)
```

- The **get.oturep** command generates a fasta-formatted sequence file containing only a representative sequence for each OTU.

```
mothur > get.oturep(phylip=example.final.phylip.dist, list=example.final.list, fasta=example.final.fasta, label=0.03)
```

- Transfer the following files from the linux platform to **windows**:

1. example.final.0.03.rep.fasta
2. example.final.groups.summary
3. example.final.shared

- The remaining actions are performed in **windows**
- Open example.final.shared in notepad++ , copy and paste in Excel, then copy and paste (transpose) into a new sheet. Insert 1 column at the start labelled Genus. Save this spreadsheet e.g. DiversityResults.xlsx
- Open the example.final.fasta file in notepad++ and remove all of the '-' symbols. This can be done using the find and replace tool, replacing '-' with a backspace.
- Download RDP Classifier files from https://sourceforge.net/projects/rdp-classifier/ and save in a logical location. Copy and paste the example.final.0.03.rep.fasta into the same file as the Classifier executable jar file
- **NOTE:** In order to run the following script you will need Java (download jre-7u60-windows-x64.exe)
- Open dos terminal (type cmd into windows search) and navigate to the location of the classifier exe jar file and the example.final.0.03.rep.fasta file

cd	To change directory within a drive
d:	To change drive (example shown here is to change to D drive)

- Run the classifier exe jar file to find the OTUs to the genus level. This creates two new files (outfile in the script)

```
java -Xmx1g -jar Classifier.jar --conf=0.5 --hier_outfile=example_unclassified_hier.txt --assign_outfile=example_unclassified_detail.txt --format=fixrank example.final.0.03.rep.fasta
```

- N.B. The '--assign_outfile=' command may need to be written as '--outputFile='
- Open example_unclassified_detail in Excel, copy and paste the genus column into the DiversityResults.xlsx genus column. As the DiversityResults.xlsx genus column was created from the subsampled sequences (OTUs) and the example_unclassified_detail file contains genera identified from ALL of the OTUs found in the samples , there will likely be more genera than OTUs. You will need to delete the genera corresponding to removed OTUs for the spreadsheet to make sense. This is not difficult as the genera and OTUs are organised numerically in Excel (smallest to largest).
- N.B. To further validate the genus IDs for each OTU the sequences from the example.final.0.03.rep.fasta file can be used to search the RDP-II sequence database (http://rdp.cme.msu.edu/)
- You now have a spreadsheet containing OTUs, genera and number of sequence reads per sample. Remove any genera appearing with less than 10 reads across the sample set.
- Consolidate OTUs to genus level.
- This spreadsheet can be used to calculate relative abundance of genera for a sample by looking at the % of sequence reads each genus holds. Barcharts can be created in Excel to visualise the data.
- Further data analysis (alpha and beta diversity, statistical differences between factors for sample groups) is performed using R statistical software using this spreadsheet and associated metadata.

References

R-Core-Team. (2013). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

Supplementary method S6

R scripts for statistical analysis of microbiota data

The following R scripts can be used in conjunction with different spreadsheets to analyse microbiota data. The format of the spreadsheets is described, along with the R commands for each analysis performed in the paper. The examples used are for the Bcc trial data.

1) R packages needed

The following R packages are needed for all the analysis and data visualisation

```
library(vegan)
library(goeveg)
library(car)
library(psych)
library(lattice)
library(FSA)
library(PMCMR)
library(ggplot2)
library(cowplot)
library(gamlss)
library(dplyr)
library(funrar)
library(NMF)
library(RColorBrewer)
library(colorspace)
library(nlme)
```

2) Read in OTU tables and metadata spreadsheets

#OTU table: Compile an excel spreadsheet with genus/OTU as column 1 without a heading. All the other columns have sample number as a heading and read numbers in rows corresponding to the genus/OTU in column 1. Save as a .txt file and read into R.

```
data<-read.table(file.choose(), header=T)
```

#Metadata table: Compile a spreadsheet with sample number as column 1. All other columns contain metadata (e.g. Patient number, Treatment group, qPCR total abundance, Shannon.diversity, Relative abundance of a key pathogen, FEV1). All columns have headings. Save as a .csv file and read into R.

```
meta_table<-read.csv(file.choose(),row.names=1,check.names=FALSE)
```

3) Calculate Shannon diversity and save as a new file

```
data_transpose<-t(data)
matrix_data<-data.matrix(data_transpose)
shannon.diversity <- diversity(matrix_data, "shannon")
shannon.results<-as.data.frame(shannon.diversity)
write.csv(shannon.results, "PATH/TO/FILE/Shannon_diversity_results.csv")
```

#Shannon diversity data: Shannon diversity values can be added into the metadata spreadsheet before reading to R for analysis.

4) NMDS ordination plot of Bray-Curtis dissimilarity values

#Format data

```
abund_table<-t(data)
meta_table$Treatment<-factor(meta_table$Treatment, levels = c("Screening",
"Start_OligoG", "End_OligoG","Start_Placebo", "End_Placebo","End"))
env.treatment<-as.factor(meta_table$Treatment)
env.treatment #check the levels of treatment
env.patient<-as.factor(meta_table$Patient)
env.patient #check the levels of patient
```

#Calculate Bray-Curtis dissimilarity values

```
BC_OTU_table <- vegdist(abund_table, method="bray", binary = FALSE)
mat_BC <- data.matrix(BC_OTU_table)
```

#NMDS ordination of Bray-Curtis dissimilarity values

```
sol<-metaMDS(abund_table,distance = "bray", k = 2, trymax = 50)
```

#Plot NMDS with ellipses around treatment groups and add in genus names (top 7% genera)

```
win.graph()
plot(sol, display="sites", type="n", main="Treatment")
points(sol, cex=1, pch=19, col=env.treatment)
ordiellipse(sol, group=env.treatment, show.groups = "Screening",
col="black", kind="sd", lwd=2)
ordiellipse(sol, group=env.treatment, show.groups = "Start_OligoG", col=
"red" , kind="sd", lwd=2)
```

```
ordiellipse(sol, group=env.treatment, show.groups = "End_OligoG", col=
"green3" , kind="sd", lwd=2)
ordiellipse(sol, group=env.treatment, show.groups = "Start_Placebo", col=
"blue" , kind="sd", lwd=2)
ordiellipse(sol, group=env.treatment, show.groups = "End_Placebo", col=
"cyan" , kind="sd", lwd=2)
ordiellipse(sol, group=env.treatment, show.groups = "End", col= "magenta" ,
kind="sd", lwd=2)
legend("topright", c("Screening", "Start_OligoG",
"End_OligoG","Start_Placebo", "End_Placebo","End"), cex=0.7, col=c
("black","red","green3","blue","cyan","magenta"), lwd=2)
top_7_pc<-ordiselect(abund_table, sol, ablim = 0.07, choices = c(1, 2),
method = "axes", freq = FALSE)
ordipointlabel(sol, display="species", select = top_7_pc, col="black",
cex=1, add = TRUE)

#Plot NMDS with ellipses around patient groups
win.graph()
env.patient
plot(sol, display="sites", type="n", main="Patient")
points(sol, cex=1, pch=19, col=env.patient)
ordiellipse(sol, group=env.patient, show.groups = "27610001", col="red",
kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610002", col=
"#FF7600" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610003", col=
"#FFEB00" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610004", col=
"#9DFF00" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610005", col=
"#27FF00" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610006", col=
"#00FF4E" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610007", col=
"#00FFC4" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610008", col=
"#00C4FF" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610009",
col="#004EFF", kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27610011", col=
"#2700FF" , kind="sd", lwd=2)
```

```
ordiellipse(sol, group=env.patient, show.groups = "27611002", col=
"#9D00FF" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27611005", col=
"#FF00EB" , kind="sd", lwd=2)
ordiellipse(sol, group=env.patient, show.groups = "27611006", col=
"#FF0076" , kind="sd", lwd=2)
```

#Permanova using adonis to determine whether groups are significantly different. Adonis cannot handle random effects (Patient) but can structure the formula ~ A + B to calculate the amount of variation explained by A (Patient) and then B (Treatment). This is not ideal as permanova can only really handle fixed effects, but the best that can be done this way.

```
adonis(mat_BC ~ Patient + Treatment, data =meta_table, permutations = 999)
```

#Check beta dispersion. Groups should have the same variance to satisfy the conditions of permanova.

```
beta <- betadisper(BC_OTU_table, meta_table$Treatment)
permutest(beta) #not significant so datasets should have same variance
beta2 <- betadisper(BC_OTU_table, meta_table$Patient)
permutest(beta2)#not significant so datasets should have same variance
```

5) Statistical analysis of paired samples from different treatment groups

#Subset the data into separate datasheets for combined (e.g. Start + End) and individual (e.g. Start) treatments

```
subsetScreen_End<-subset(meta_table, Treatment=="Screening"|
Treatment=="End")
subsetOligoGstart_end<-subset(meta_table, Treatment=="Start_OligoG"|
Treatment=="End_OligoG")
subsetPlacebostart_end<-subset(meta_table, Treatment=="Start_Placebo"|
Treatment=="End_Placebo")

subsetScreen<-subset(meta_table, Treatment=="Screening")
subsetEnd<-subset(meta_table, Treatment=="End")
subsetStartOligoG<-subset(meta_table, Treatment=="Start_OligoG")
subsetEndOligoG<-subset(meta_table, Treatment=="End_OligoG")
subsetStartPlacebo<-subset(meta_table, Treatment=="Start_Placebo")
subsetEndPlacebo<-subset(meta_table, Treatment=="End_Placebo")
```

#Statistical analysis of differences in Shannon diversity/total abundance by qPCR between treatment groups using Wilcoxon signed-rank tests (example given for Screen

- End group and Shannon diversity, but same process applies to other treatment group comparisons and qPCR data)

#Check normality (shapiro.wilk), the spread of the data (boxplots) and homogeneity of variances (bartlett,test and leveneTest)

```
by (subsetScreen_End$shannon.diversity, subsetScreen_End$Treatment,
shapiro.test)
boxplot(subsetScreen_End$shannon.diversity~subsetScreen_End$Treatment)
bartlett.test(subsetScreen_End$shannon.diversity~subsetScreen_End$Treatment
)
leveneTest(subsetScreen_End$shannon.diversity~subsetScreen_End$Treatment)
```

#Further investigation of each treatment group individually for normality (histogram of data distribution and qqplot)

```
hist(subsetScreen$shannon.diversity)
qqnorm(subsetScreen$shannon.diversity)
qqline(subsetScreen$shannon.diversity)
```

#In this case data were normal with homogeneous variances so used a Wilcoxon signed-rank test

```
wilcox.test(subsetScreen$shannon.diversity, subsetEnd$shannon.diversity,
paired = T)
```

#Statistical analysis of differences in relative abundance of a key pathogen between treatment groups using GAMLSS-BEINF and (mu) logit links (example given for Screen - End group).

#Check the distribution of the data, here the data follows a zero-one inflated beta distribution for both treatment groups being compared

```
hist(subsetScreen$Burkholderia)
hist(subsetEnd$Burkholderia)
```

#Perform the GAMLSS-BEINF analysis

```
subsetScreen_End$Burkholderia_proportion<-subsetScreen_End$Burkholderia/100
#make proportion and append column to dataset
subsetScreen_End$Patient<-as.factor(subsetScreen_End$Patient)
modBcc <- gamlss(Burkholderia_proportion ~ Treatment + random(Patient),
family = BEINF, data = subsetScreen_End)
```

```
summary(modBcc)
```

#Plotting the results of all the comparisons as boxplots. There are three pairs of treatment comparisons (Screen – End, OligoG Start – OligoG End and Placebo Start – Placebo End) for three response variables (Shannon diversity, qPCR total abundance and relative abundance) giving nine plots.

#Plotting the nine plots individually

```
win.graph()
plot1<-ggplot(subsetScreen_End, aes(x = Treatment, y = shannon.diversity,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,2.5)
plot2<-ggplot(subsetOligoGstart_end, aes(x = Treatment, y =
shannon.diversity, fill=Treatment)) + geom_boxplot() + theme_bw() +
scale_fill_brewer(palette = "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,2.5)
plot3<-ggplot(subsetPlacebostart_end, aes(x = Treatment, y =
shannon.diversity, fill=Treatment)) + geom_boxplot() + theme_bw() +
scale_fill_brewer(palette = "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,2.5)
plot4<-ggplot(subsetScreen_End, aes(x = Treatment, y = qPCR,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,9)
plot5<-ggplot(subsetOligoGstart_end, aes(x = Treatment, y = qPCR,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,9)
plot6<-ggplot(subsetPlacebostart_end, aes(x = Treatment, y = qPCR,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,9)
plot7<-ggplot(subsetScreen_End, aes(x = Treatment, y = Burkholderia,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
```

```
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,100)
plot8<-ggplot(subsetOligoGstart_end, aes(x = Treatment, y = Burkholderia,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,100)
plot9<-ggplot(subsetPlacebostart_end, aes(x = Treatment, y = Burkholderia,
fill=Treatment)) + geom_boxplot() + theme_bw() + scale_fill_brewer(palette
= "Set1") + theme(legend.position = "none") +
theme(axis.title.x=element_blank()) + theme(axis.title.y=element_blank()) +
ylim (0,100)
```

#Combining all nine plots into one plot

```
plot_grid(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9,
nrow = 3, align = "h")
```

6) Linear regression FEV1 and Shannon diversity

#In the Bcc trial 'End' samples did not have associated FEV1 data, subset the data to include all time points except 'End'.

```
subsetFEV1_data<-subset(meta_table, Treatment=="Screening"|
Treatment=="Start_OligoG"| Treatment=="End_OligoG"|
Treatment=="Start_Placebo"| Treatment=="End_Placebo")
```

#Plot the two variables FEV1%predicted and Shannon diversity and perform linear regression to plot a trend line. This steps allows visualisation of the data and identification of potential correlations.

```
linearmod<-lm(subsetFEV1_data$shannon.diversity~
subsetFEV1_data$FEV1_percent_predicted)

print(linearmod)

summary(linearmod)

plot(subsetFEV1_data $FEV1_percent_predicted,
subsetFEV1_data$shannon.diversity, pch=16, col="black", main="Bcc FEV1 vs.
diversity")

abline(lm(subsetFEV1_data$shannon.diversity~
subsetFEV1_data$FEV1_percent_predicted))
```

#Patient needs to be taken into account as a random effect to determine statistical significance. Use a mixed model to do this (nlme package).

```
mixedlm<-lme(shannon.diversity~FEV1_percent_predicted, random = ~1|Patient,
data=subsetFEV1_data)
summary(mixedlm)
```

#Check residuals of model, residuals need to be evenly distributed around 0 and to fall along the line in a qqplot.

```
plot(mixedlm)
qqnorm(resid(mixedlm))
qqline(resid(mixedlm))
```

7) Analysis of V1 samples: hierarchical clustering

#For V1 Screening samples hierarchical clustering was performed for the Bcc trial and the *P. aeruginosa* trial. NMDS ordination of Bray-Curtis dissimilarity values and boxplots of qPCR total abundance values were also used to investigate the data for the Bcc trial. As the R scripts for NMDS and plotting boxplots have been described previously they will not be shown again. Here hierarchical clustering of the Bcc V1 data is shown.

#Read in OTU tables and metadata spreadsheets, the OTU table and metadata spreadsheets used previously need to be modified to remove all samples except those at screening (V1). This can be done in excel and the files read into R as before.

```
data_V1<-read.table(file.choose(), header=T)
meta_table_V1<-read.csv(file.choose(),row.names=1,check.names=FALSE)
```

#Format the OTU table to find the relative abundance of each genus in each sample.

```
OTU_total<-cbind(data_V1, total = rowSums(OTU_table_to_format))
OTU_order<-OTU_total[order(-OTU_total$total),]
OTU_order_no_total<-subset(OTU_order, select = -c(total))
OTU_order_no_total_matrix<-as.matrix(OTU_order_no_total)
transposed_matrix<-t(OTU_order_no_total_matrix)
OTU_relabund<-make_relative(transposed_matrix)
transposed_relabund<-t(OTU_relabund)
percent_relabund<-(transposed_relabund)*100
```

#Indicate which column annotations you need for the heatmap. Here the *Burkholderia* species is the column annotation (in the column 'Species' of the metadata file)

```
env.species<-as.factor(meta_table_V1$Species)

env.species
```

#Choose the colour palette for the heatmap. The first command opens a colour palette so you can choose the colour scale, the second command reverses the colours in the palette (light-dark or dark-light)

```
mypalette<-choose_palette()

col_rev <- rev(mypalette(25))
```

#Draw the heatmap. These commands specify to only include the top 15 rows by highest OTU abundance. Hierarchical clustering is performed using Bray-Curtis dissimilarity values and Ward's clustering algorithm.

```
win.graph()

Top15_rownames<-rownames(percent_relabund[1:15,])

heatmap_Bcc_V1=aheatmap(percent_relabund, distfun = function(x) vegdist(x,
method = "bray"), hclustfun = function(x) hclust(x, method = "ward.D2"),
color = col_rev, treeheight = 30, Rowv=NA, annCol = env.species, annColors
= list(c("B. cenocepacia" ="#b2df8a", "B. multivorans"="#fdbf6f","No Bcc
detected"="#fb9a99")), subsetRow =  Top15_rownames)

heatmap_Bcc_V1
```

8) **Pearson product-moment correlation coefficient (PPMCC) using the pearson correlation to compare bacterial community composition between paired samples**

#OTU table: Compile an excel spreadsheet for each patient with genus/OTU as column 1 without a heading. All the other columns have sample number as a heading and read numbers in rows corresponding to the genus/OTU in column 1. Save as a .txt file and read into R. An example for one patient is given below.

```
Patient1<-read.table(file.choose(), header=T, check.names="FALSE")
```

#Perform a correlation test for all of the different combinations of paired samples for each patient. An example for one patient is given below.

```
cor.test(Patient1$V1_1, Patient1$V1_2, method = "pearson")
cor.test(Patient1$V2_1, Patient1$V2_2, method = "pearson")
cor.test(Patient1$V4_1, Patient1$V4_2, method = "pearson")
cor.test(Patient1$V5_1, Patient1$V5_2, method = "pearson")
cor.test(Patient1$V7_1, Patient1$V7_2, method = "pearson")
cor.test(Patient1$V8_1, Patient1$V8_2, method = "pearson")
```

#These values can be collated to determine key statistics such as mean and median, and the spread of the data visualised using boxplots