Effects of an experimentally evolved defensive microbe on its host-microbiome system



Dylan Dahan

Department of Zoology

Merton College

University of Oxford

A thesis submitted for the degree of

Master of Science by Research

Trinity 2017

Table of Contents

List of Figures 3

List of Tables 3

Acknowledgements 4

Abstract 6

Introduction 7

- Symbionts protective effects on hosts and associated signatures 7
- Symbionts effects on the microbiome 8
- C. elegans as a model for studying host-microbe interactions 9
- Influence of defensive microbes on C. elegans host-microbiome system 11

Results 14

- C. elegans exposure to E. faecalis OG1RF 14
- DEGs under E. faecalis SE and E. faecalis CCE exposures 16
- Pathogen specificity with E. faecalis SE and E. faecalis CCE exposures 17
- GO terms functionally enriched with E. faecalis SE and E. faecalis CCE exposures 22
- Processing of 16S rRNA reads from C. elegans' natural microbiota 28
- Effects of pre-exposure on microbiota diversity 30
- Differentially abundant microbiota influenced by pre-exposure treatments 33
- C. elegans transcript correlations with Enterococcus abundance 36
- Evolved E. faecalis colonization efficacy and protection persistence 37
- C. elegans transcript correlations with E. faecalis colonization efficacy 39

Discussion 41

- E. faecalis CCE effects on C. elegans 41
- E. faecalis CCE effects on the C. elegans microbiome 46
- Future directions 49

Methods 50

- Strains 50
- C. elegans exposures to E. coli OP50 and treatments 50
- RNA extraction and library preparation 51
- Compost preparation 52
- Worm compost exposure and harvesting 53
- DNA extractions 53
- 16S rRNA library preparation 53
- Gut accumulation enumeration and protection persistence 54
- RNASeq bioinformatic processing and analyses 55
- Code availability 57

Supplementary Figures 58

Supplementary Tables 66

Bibliography 68

Supplementary Files 77

- Supplementary file 1. R Markdown file outlining gut enumeration and protection analyses 77
- Supplementary file 2. Snakemake commands for processing RNA reads with Trimmomatic and kallisto. 83

Supplementary file 3. R Markdown file outlining differential expression and GO term analysis. 84

List of Figures

Figure 1. C. elegans significant DEGs under E. faecalis SE and E. faecalis CCE exposure. 17

Figure 2. Pathogen specific and common genes significantly differentially expressed in *C. elegans* under *E. faecalis* SE and *E. faecalis* CCE exposures. 19

Figure 3. GO term analysis of significant DEGs from comparing *C. elegans* under *E. faecalis* SE and *E. faecalis* CCE exposures to *E. faecalis* Anc exposure 22

Figure 4. GO term analysis of significant DEGs from comparing C. elegans under E. faecalis CCE exposure to E. faecalis SE exposure 25

Figure 5. Alpha diversity measurements of *C. elegans* microbiota after compost exposure 30

Figure 6. Principal coordinate analyses (PCoA) on weighted UniFrac scores of *C. elegans* microbiota 31

Figure 7. RSVs that significantly differ in abundance in *C. elegans* microbiota after different pre-exposure treatments and compost exposure 33

Figure 8. Evolved *E. faecalis* strains colonization in *C. elegans* and effects on *S. aureus* induced mortality amongst natural microbiome 37

Figure 9. Correlation between *E. faecalis* CFUs in *C. elegans* guts and *ilys-3* TPM values 39

List of Tables

Table 1. Thesis predictions, support and approaches 12

Table 2. *clec* gene β-values 20

Table 3. DEGs from *E. faecalis* CCE to *E. faecalis* SE comaprison mapping to enriched GO terms 26

Table 4. Alpha diversity measurements of *C. elegans* microbiota after compost exposure 28

Acknowledgements

The thesis is the last piece of scientific literature that is often only submitted with a single author, and that's strange. This project owes its strides to the combined efforts and conversation of lab-mates, collaborators, and pub-goers. My friends in the Interdisciplinary Bioscience DTP, Michael Niklaus, Susanna Streubel, Dante Wasmuht, Xiqui Bach Pagés, and Sam Watson, you were a distinct pleasure to scratch heads with, from linear algebra to cellular automata. My friends in the Aboobaker lab, from the pre-asbestos days, thanks for sharing the ease at which you conduct molecular assays and preps. Damian Kao, my good friend, and computational mentor thank you for sharing your bioinformatic wizardry and guiding me in our many analyses. My new friends in the Hodgkin and Woollard labs, thank you for sharing your lab spaces and making us in the King Lab feel at home.

My dearest friends in the King Lab, Alex Betts, Suzie Ford, Alice Ekroth, Anke Kloock, Mariá Ordovás-Montañés, Charlotte Rafaluk-Mohr, and Jordan Sealey, you have been too much fun but nonetheless showed me that fun doesn't come at the cost of being prolific. I am so fortunate to have done my thesis with such a lovely group of people. My co-supervisor, Gail Preston, thank you for your guidance and helping me solve problems with your vast knowledge of many systems, from *C. elegans* to soil microbes. And, my dearest thanks my supervisor, Kayla King, for your continuous scientific and personal support, and for turning my vague thoughts on microbiome literature into clear ecological and evolutionary hypotheses.

Abstract

Dylan Dahan
Merton College
An abstract submitted for the degree of M.Sc. by Research
Trinity 2017

Defensive microbes readily influence hosts and their microbiome. And, since hosts and their microbiome are not disparate but comprise an integrated host-microbiome system, it follows that defensive microbes should alter the system as a whole. Nonetheless, direct evidence on how defensive microbes influence host-microbiome systems is lacking. Using *C. elegans* and their natural microbiome as a model host-microbiome system and an experimentally evolved defensive *E. faecalis* strain, I integrated host RNA sequencing, microbiome 16S rRNA sequencing, and phenotypic assays to show effects of a defensive microbe on its host-microbiome system. My results indicate that a defensive microbe can substantially alter its host transcriptome while influencing little change on its host's microbiome. Additionally, that a defensive microbe can colonize better than its non-defensive counterparts and maintain protective effects even amongst a natural microbiome. This thesis reveals some outcomes and utility of defensive microbes that can be translated to both natural and applied contexts. Additionally, this thesis promotes experimental evolution as a key tool in investigating evolutionary and ecological outcomes of symbiosis.

Abbreviations:
Anc: ancestor; CCE: co-colonized evolved; CFUs: colony-forming units; CTLD: C-type lectin-like domain; DEG: Differential expressed gene; GIT: Gastrointestinal tract; GO: Gene ontology; LB: lysogeny broth; MAPK: Mitogen activated protein kinase pathway; PCoA: Principle coordinate analysis; RSV: Ribosomal sequence variant; SE: single evolved; THB: Todd Hewitt broth; TSA: tryptic soy agar

Introduction

Hosts and their complex microbial communities (i.e., microbiomes) are intimately intertwined. Individual symbionts play fundamental roles in affecting overall host physiology through interactions with both parties of the host-microbiome system. Symbionts are any organism that shares an evolutionary history with their host, ranging from mutualists, which confer and receive a benefit, to pathogens, which receive a benefit but have adverse effects on their host. Further, symbiont roles are not mutually exclusive and an organism that is a mutualist in one context may be a pathogen in another, such is often the case with net mutualists (Dethlefsen et al. 2007; King et al. 2016). Symbionts can affect host development (Hosokawa 2016; Shin et al. 2011), speciation (Baumann et al. 1995), nutrient acquisition (Rubino et al. 2017), immune maturation (Chung et al. 2012; Cosseau et al. 2008), and pathogen susceptibility (Sorg & Sonenshein 2008; Abt & Artis 2013), and can also affect host microbiomes by influencing the assembly of other microbes (Schwarz et al. 2016) and contributing to available microbial gene pools (Stecher et al. 2012). Symbiont influences are not exclusive, i.e., either host or microbiome altering, but can be integrated. For instance, early monocolonization by a bacterium can increase host pathogen load upon natural microbiome exposure, and thus influence detrimental effects on host development (Schwarz et al. 2016). Understanding how a symbiont affects a host organism necessitates understanding how it influences the whole host-microbiome system.

Symbionts protective effects on hosts and associated signatures

Multicellular hosts harbor diverse microbiomes that provide a range of benefits, particularly including protection against pathogens (Bäumler & Sperandio 2016; Ford et al. 2016). Resident protective microbiome members, called defensive microbes, exist in nature (Hrček et al. 2016; Oliver et al. 2013; Parker et al. 2013) and are important for applied contexts (Sorg & Sonenshein 2008; Becker et al. 2009; Nakatsuji et al. 2017), such as mitigating infection (Fuentes et al. 2014) and preventing disease transmission in humans (Walker et al. 2011). In interfering with pathogens, defensive microbes can beneficially contribute to or alter the host metabolome (Marcobal et al. 2013), prime the host immune system (Cosseau et al. 2008), provide colonization-mediated resistance (Buffie & Pamer 2013), or promote overall homeostasis at infection sites (Park et al. 2016). These modes of protection vary and so do their signatures. For example, signatures underlying colonization-mediated resistance can involve suppression of symbiont-related inflammatory genes (Abt & Artis 2013; Cosseau et al. 2008); those underlying immune priming involve stimulating pathogen specific transcriptional pathways to basal levels (Montalvo-Katz et al. 2013); and those underlying homeostasis in the host gastrointestinal tract involve stimulating epithelial cell turnover and propagation (Park et al. 2016; Cosseau et al. 2008; van Baarlen et al. 2011). Exploring these signatures reveals mechanisms of protection and thus offers key insights into how defensive microbes modulate host physiology.

Symbionts effects on the microbiome

Symbionts can alter microbiomes in several ways. Beneficial ways include specifically limiting success of pathogens and selectively excluding nonsymbionts

(Kremer et al. 2013). Adverse affects also exist and include increasing the colonization rates of other pathogens (Schwarz et al. 2016) and contributing mobile genetic elements, such as plasmids containing virulence or resistance mechanisms, to microbial gene pools (Stecher et al. 2012). In addition, abiotic factors can have ecological and evolutionary influences on microbiomes (Hall et al. 2016) (Gomez & Buckling 2011). While these modes of microbiome alteration are known, there is sparse evidence on the extent to which individual symbionts shape the composition of other mutualistic constituents of host microbiomes, such as core (i.e., essential microbes found in the majority of a species microbiomes) microbiome members. The possibility of such adverse alterations to core microbiome members is not improbable, since defensive symbionts can offer protection against pathogens via metabolites, such as superoxide antimicrobials (King et al. 2016), violacein (Brucker et al. 2008) and deoxycholate (Sorg & Sonenshein 2008), which are not necessarily species specific (Broxton & Culotta 2016). Pros, such as efficacy in preventing infections, and cons, such as increasing other pathogen susceptibilities, taken together, it is necessary to investigate the utility but also the consequences on microbiomes under exposure to defensive microbes.

C. elegans as a model for studying host-microbe interactions

C. elegans is a supermodel for biology, including for the study of natural and lab-developed host-microbe interactions (Clark & Hodgkin 2013; Cabreiro & Gems 2013; Petersen et al. 2015). It's genome was sequenced earlier than any metazoan, in 1998 (C. elegans Sequencing Consortium 1998), it's complete cellular pathways have been mapped (Sulston & Horvitz 1977), and there are publicly available, and

maintained genomic, transcriptomic and proteomic *C. elegans* databases (Howe et al. 2016). These nematodes are also bacteriovores, and thus continually and directly sample their surrounding bacterial environments (Félix & Braendle 2010). Their gastrointestinal tract is continually exposed to surrounding microbes and can be co-colonized by pathogens and commensals (Peleg et al. 2008; Niu et al. 2016; Montalvo-Katz et al. 2013). They are easily reared in a gnotobiotic setting, *sans* intensive gnotobiotic procedures, allowing for controlled assembly of diverse microbiota in their gastrointestinal tract (King et al. 2016; Portal-Celhay & Blaser 2012). And, they are lab tractable and have large population sizes. Many of these attributes also make *C. elegans* a suitable model for studying the microbiome (As discussed in Zhang et al. 2017). Indeed a recent collection of seminal studies (Dirksen et al. 2016; Berg et al. 2016; Samuel et al. 2016) revealed a conserved, core microbiome for *C. elegans* from diverse environments (e.g., natural soil and lab microcosms). Further, the *C. elegans* microbiome, similar to humans and more complex models (Fritz et al. 2013), is comprised of diverse bacterial commensals that play fundamental roles in maintaining host physiology (Samuel et al. 2016).

Some bacterial roles in *C. elegans* have even been specifically investigated in terms of innate immunity and associated transcriptional responses (Irazoqui et al., 2010; Wong et al., 2007; Montalvo-Katz et al. 2013). This includes a key example of a naturally isolated defensive symbiont, *Pseudomonas mendocina*, which protects these nematodes from *Pseudomonas aeruginosa* infection through priming of the P38 mitogen activated protein kinase pathway (MAPK) (Montalvo-Katz et al. 2013).

C. elegans are also useful for studying the evolution of host-microbe interactions (Schulte et al. 2011; Morran et al. 2016; King et al. 2016; Discussed in

Gray & Cutter 2014). This primarily owes to their lab tractability and relatively short generation times (~4 days). *C. elegans* evolution studies have so far focused on host-pathogen coevolution (Morran et al. 2011) and mating system evolution (LaMunyon et al. 2006), but King et al. (2016) recently used this model host to study the *in vivo* evolution of defensive microbes. Taken together, with *C. elegans* as an established model for studying host-microbe interactions, it's well-defined core microbiomes, prior work on their microbe-mediated immune responses, and utility in experimental evolution, it is in a prime position to be utilized as a model host-microbiome system.

Influence of defensive microbes on C. elegans host-microbiome system

Here, I aim to describe the influences of an experimentally evolved defensive microbe on the *C. elegans* host-microbiome system. I assay how the defensive microbe influences the host transcriptome, shapes the assembly of the host's natural microbiota, colonizes the host, and sustains protection in the context of natural microbiota exposure. I use King et al.'s (2016) experimentally evolved *Enteroccocus faecalis*, which was evolved *in vivo* to suppress *Staphylococcus aureus* infection, thereby defending *C. elegans* against infection-induced mortality (King et al. 2016). The ancestor *E. faecalis* was originally isolated from the human gastrointestinal tract (Garsin et al. 2001). The evolved *E. faecalis* is a symbiont but a net mutualist, by substantially reducing mortality caused by *S. aureus* from 60% to <1% but nonetheless remaining costly in the absence of *S. aureus* (King et al. 2016). Also, this protective *E. faecalis* directly inhibits *in vitro* growth of *S. aureus* through the production of superoxides, a reactive oxygen anion that can induce growth

restraints via oxidative stress (King et al. 2016). This defensive symbiont is called *E. faecalis* CCE (for co-colonized evolved), since it was evolve *in vivo* with co-colonization by *S. aureus*. As a control for an *in vivo* evolved symbiont without the selective pressure for protection, I use a non-protective strain of *E. faecalis* that was evolved *in vivo* without the presence of *S. aureus*, called *E. faecalis* SE (for single evolved). As a control for a non-defensive microbe that does not have a shared evolutionary history with *C. elegans*, I use the non-protective ancestor strain, *E. faecalis* Anc (for ancestral). With these evolutionary controls, I can resolve some of the evolutionary outcomes of defensive symbiosis on a host-microbe system. To explore defensive microbe influences on host signatures that underlie protection and mutualism, I used RNA sequencing (RNASeq) to investigate how *E. faecalis* CCE influences the host transcriptome and alters transcriptional signatures indicative of microbe-mediated protection and colonization. To assess how *E. faecalis* CCE shapes the assembly of natural microbiota and to explore possible consequences on microbiota assembly, such as increased pathogen susceptibility, I conducted 16S rRNA sequencing on *C. elegans* early exposed to monocultures of microbes after *C. elegans* have been exposed to natural microbial communities in compost. To investigate if increased colonization resulted as an outcome of symbiont evolution, I used a standard gastrointestinal tract bacterial enumeration assay. Lastly, to see if *E. faecalis* CCE persists to protect amongst a natural *C. elegans* microbiome, I exposed *C. elegans* to *S. aureus* after compost exposure. Broadly, I aim to provide a more detailed view of the evolved influences of a defensive symbiont on its host-microbiome system by testing the predictions in Table 1.

Table 1 Thesis predictions, support and approaches.

Prediction	Support	Approach
E. faecalis strains (Anc, SE and CCE) will have distinguishable host transcriptional effects.	Natural *C. elegans* symbionts with strain level variation differently influence host physiology (Samuel et al. 2016) and strain level variation drives niche specialization in other host microbiomes (Rubino et al., 2017) and transcriptional signatures in other systems (Mandel et al. 2009).	Compare DEGs and related gene ontology (GO) terms between *E. faecalis* CCE, *E. faecalis* SE and *E. faecalis* Anc
E. faecalis CCE will influence differential expression of *C. elegans* genes related to oxidation-reduction processes.	*E. faecalis* CCE produces superoxides *in vitro* (King et al. 2016). The *C. elegans* transcriptome is readily modified by the presence of oxidative species (McCallum & Garson 2016).	Query *E. faecalis* CCE transcriptome comparisons for DEGs and GO terms related to oxidation-reduction processes.
E. faecalis CCE will upregulate *C. elegans* genes associated with *S. aureus* infection.	Microbe-mediated protection in *C. elegans* can be associated with basal stimulation of specific pathogen-associated gene pathways (Montalvo-Katz et al. 2013).	Query *E. faecalis* CCE transcriptome comparisons for DEGs previously reported as *S. aureus* infection biomarkers related to defense (Irazoqui et al. 2009)
E. faecalis CCE will downregulate genes associated with *E. faecalis* infection.	Microbe-mediated protection can be associated with downregulation of symbiont-specific inflammatory responses (Cosseau et al. 2008).	Query *E. faecalis* CCE transcriptome comparisons for DEGs previously reported associated with *E. faecalis* infection (Wong et al. 2007)
E. faecalis CCE, *P. mendocina* and my non exposure control (E. coli OP50) will differently shape C. elegans microbiome assembly. Specifically, *E. faecalis* exposure will result in higher *E. faecalis* colonization and *P. mendocina* exposure will not limit *E. faecalis* colonization.	*E. faecalis* CCE outcompetes *S. aureus* (King et al. 2016). *P. mendocina* prevents *P. aeruginosa* colonization (Montalvo-Katz et al., 2013) and does not limit *E. faecalis* (Montalvo-Katz et al. 2013). Early exposure to *E. coli* OP50 allows for natural microbiota assembly (Berg et al., 2016; Dirksen et al., 2016).	Compare microbiomes of *C. elegans* after treatment and compost exposure, specifically comparing microbiota alpha diversity, beta diversity, and differential abundance of microbial genera.
In vivo evolution of *E. faecalis* CCE and *E. faecalis* SE will result in increased gut colonization in *C. elegans*.	Increased colonization is postulated ro result from in vivo symbiont evolution (Hoang, Morran, & Gerardo, 2016) and has shown to occur in *E. coli* serial passaged across *C. elegans* (Portal-Celhay & Blaser, 2012).	Enumerate colony-forming units (CFUs) of *E. faecalis* strains in *C. elegans* after early exposures.
Protective effects of *E. faecalis* CCE will remain amongst a natural microbiome be less effective at protecting overall.	Fitness constraints imparted by diverse interactions in polymicrobial communities can change (Gomez & Buckling, 2011) and even dilute (Sivan et al. 2015; Lenhart & White 2017) phenotypes normally observed in reduced systems.	Conduct mortality assay of *E. faecalis* treated *C. elegans* on *S. aureus* after they have been exposed to compost.

Results

C. elegans *exposure to* E. faecalis *OG1RF*

Pathogenesis can be better understood by using RNASeq to explore the mechanisms underlying microbial associated-molecular patterns. Signatures can be associated with numerous *C. elegans* pathogens, including as *E. faecalis* (Wong et al. 2007). To better describe *E. faecalis*-associated signatures in *C. elegans* I used RNASeq to compare genes expressed by *C. elegans* upon *E. faecalis* OG1RF exposure to genes expressed by standard *E. coli* OP50 exposure. Four *C. elegans* populations were exposed at the L3/L4 stage to *E. faecalis* OG1RF or *E. coli* OP50 until young adults (24h), then RNA was extracted, transcripts quantified, and expression of genes compared. Comparisons were only between *C. elegans'* RNA from the different treatments. I used these conditions to match the conditions previously used to describe test *E. faecalis* OG1RF protective effects against *S. aureus* in *C. elegans* (King et al. 2016). *C. elegans* exposure to *E. faecalis* induced significant differential expression of 16653 transcripts compared to the *C. elegans* control (*E. coli* OP50 exposure) (Supplementary table 1; adj-P < 0.05; Wald-test). These mapped to 4,840 unique differentially expressed genes (DEGs), as defined by unique WormBase IDs (adj-P < 0.05; β-value > 1). Comparing our results to the other study analyzing *C. elegans* exposure to *E. faecalis* OG1RF (Wong et al. 2007), I observed 65.34% overlap of DEGs. Discrepancies are likely due to experiment-specific culture and maintenance conditions; these include that the *C. elegans* in our study were exposed to *E. faecalis* at the L3/L4 larval stage while those in Wong et al. (2007) were exposed at the mid-L4 stage (Boeck et al. 2016), that I cultured *E. faecalis* OG1RF in

Todd Hewitt broth (THB) and Wong et al. (2007) cultured *E. faecalis* OG1RF in brain-heart infusion broth, that I cultured *E. faecalis* OG1RF and *E. coli* OP50 overnight at 30°C and Wong et al. (2007) cultured *E. faecalis* OG1RF and *E. coli* OP50 overnight at 37°C, and that I plated *E. faecalis* OG1RF for *C. elegans* exposure on tryptic soy agar (TSA) plates while Wong et al. (2007) plated *E. faecalis* OG1RF for *C. elegans* exposure on nematode growth medium (NGM).

To demonstrate experimental equivalence and strengthen the case for DEGs associated with *E. faecalis* exposure, I highlight 11 previously described DEGs related to general pathogenesis and *E. faecalis* exposure, including three genes encoding aspartyl proteases (*asp* genes) and three C-type lectin-like domain (CTLD) genes (*clec* genes) that were confirmed by quantitative real-time quantitative PCR (RTqPCR) and several others that increased according to microarray analysis (Wong et al. 2007). My results indicate that 10/11 of these RTqPCR genes agreed with previous results by increasing or decreasing in expression similarly upon *E. faecalis* exposure (Supplementary Figure 1), with the exception being *npp-13* marginally decreasing in expression rather than increasing. Only directional but not magnitude comparisons are applicable since I indicate expression differences using β-value measurements inherent to sleuth's differential expression analysis – where β-values correspond to an effect size in log-transformed space, and Wong et al. (2007) used log2 fold change, which are log transformed values from previously untransformed values.

Further corroborating *E. faecalis* specific immune regulation, I observed concordant upregulation of a number CTLD and lysozyme genes previously associated with *E. faecalis* infection (Pees et al. 2016; Schulenburg et al. 2008; Wong

et al. 2007). In specific, these include 5/8 *E. faecalis* associated lysozyme genes *(lys-7*, *lys-10*, *spp-8*, *lys-4* and *lys-5*) and 8/12 *E. faecalis* associated *clec* genes (*clec-67*, F40F4.6, T25C12.3, *clec-63*, *clec-65*, *clec-47, clec-54* and *clec-67*). Again, some discrepancies are likely due to different assay conditions.

C. elegans exposure to *E. faecalis* functionally enriched 62 gene ontology (GO) terms, with a fold enrichment ranging from 1.07-1.50 and an average of 1.24 ± s.e. 0.01 (Supplementary Figure 2). GO terms with the most genes mapping to them include embryo development ending in birth or egg hatching (GO:0009792; 2561 genes); reproduction (GO:0000003; 1860 genes); and nematode larval development (GO:0002119; 1640 genes). Of the 16652 significantly differentially expressed genes, 7326 mapped to GO terms.

DEGs under E. faecalis SE and E. faecalis CCE exposures

I next sought to describe how exposure to *E. faecalis* SE and *E. faecalis* CCE regulates differential gene expression in *C. elegans* compared to exposure with *E. faecalis* Anc. To do so, I compared RNA profiles from *C. elegans* exposed to *E. faecalis* SE or *E. faecalis* CCE to those exposed to *E. faecalis* Anc, with four replicate populations per treatment. Again, with transcripts I quantified differential expression of genes and functional enrichment of GO terms. This allowed us to describe how our *in vivo* evolved *E. faecalis* regulated the *C. elegans* transcriptome different than their ancestor. I found there were 135 DEGs from the *E. faecalis* SE exposure and 458 DEGs from the *E. faecalis* CCE exposure compared to *E. faecalis* Anc (Wald-test; adj-P <0.05; DEG list in Supplementary Table 2), 45 DEGs of which were shared. I highlighted the top 75 DEGs from both comparisons and those that

were shared (Figure 1abc). Of these, the average absolute change in expression for DEGs from the *E. faecalis* SE treatment was 0.76 (± s.e. 0.12) and from the *E. faecalis* CCE treatment was 2.2 (± s.e. 0.16). DEGs from *E. faecalis* CCE treatment induced on average 3x greater changes in expression than the *E. faecalis* SE treatment, a finding that was significant (Mann-Whitney test; $P < 0.01$).

Expression changes from 45-shared DEGs between *E. faecalis* SE and *E. faecalis* CCE were not different but in fact highly correlated (Pearson's; $R = 0.966$; $T = 24.5$; $df = 43$; $P << 0.01$; Figure 1c). Further, *E. faecalis* SE and *E. faecalis* CCE exposures shared functional enrichment of three GO terms, collagen trimer (GO:0005581), structural constituent of the cuticle (GO:0042302), and extracellular region (GO:0005576) (Figure 2a). Several of the shared DEGs, *dpy* genes that encode external collagen of the cuticle (Wheeler & Thomas 2006; Taffoni & Pujol 2015), mapped to the collagen and cuticle terms. In all, the *E. faecalis* CCE treatment induced differential expression of 3.4x more genes than the *E. faecalis* SE treatment and on average induced greater differential expression, but when DEGs were shared the treatments induced similar expression differences and functions related to collagen and cuticle.

Pathogen specificity with E. faecalis SE and E. faecalis CCE exposures

I next sought to describe how *C. elegans* responses to *E. faecalis* SE and *E. faecalis* CCE exposures are related to specific pathogen signatures. Specifically, I was interested to see *E. faecalis* SE and *E. faecalis* CCE induced differential expression of *E. faecalis* specific genes, *E. faecalis* and *S. aureus* common genes, and *S. aureus*-

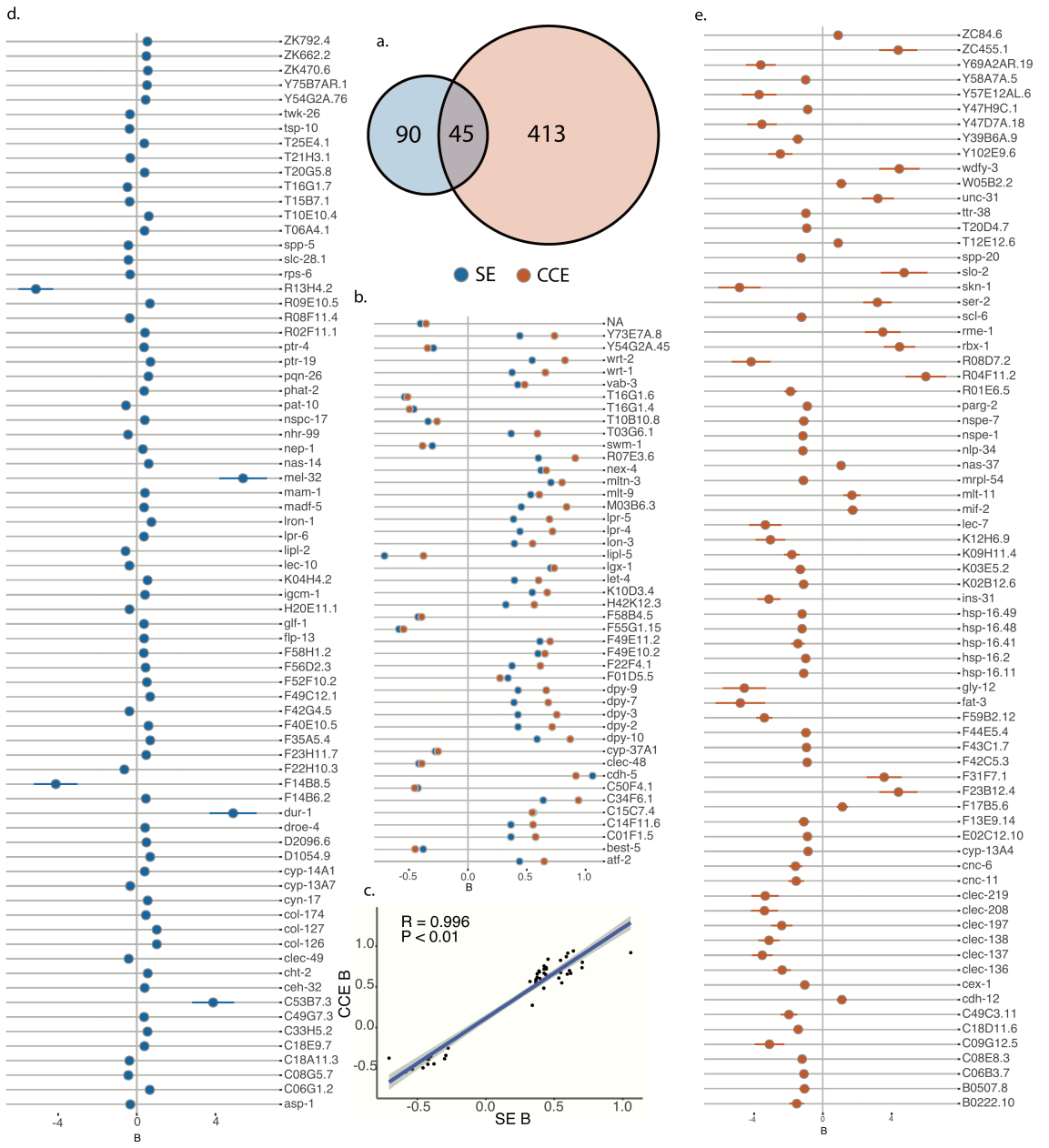


Figure 1. *C. elegans* significant DEGs under E. faecalis SE and E. faecalis CCE exposure. Significant DEGs from *C. elegans* exposed *E. faecalis* SE or *E. faecalis* CCE compared to exposure with *E. faecalis* Anc. **a.** Venn diagram showing sets of significant DEGs from *C. elegans* exposed to *E. faecalis* SE (orange), *E. faecalis* CCE (blue) or DEGs in the intersection. **b.** All 45 significant DEGs from *C. elegans* exposed to *E. faecalis* SE or *E. faecalis* CCE with x-axis showing β-values. **c.** Scatterplot mapping β values of matching significant DEGs from C. elegans exposed to *E. faecalis* CCE or *E. faecalis* SE (Pearson's; R=0.966; T = 24.5; df = 43; P << 0.01). **d.** Top 75 significant DEGs in *C. elegans* exposed to *E. faecalis* SE. **e.** Top 75 significant DEGs from *C. elegans* exposed to evolved *E. faecalis* CCE. x-axes are again β-values from Wald-Test. FDR adj-P < 0.05; sleuth. Error bars ± s.e.

specific genes. I used previous data (Wong et al. 2007; Irazoqui et al. 2008) to compile lists of *E. faecalis* and *S. aureus* common and specific genes, and queried our *E. faecalis* SE and *E. faecalis* CCE exposure DEGs for these.

E. faecalis SE induced differential expression of four *E. faecalis* specific genes and one *S. aureus*-specific DEG (Figure 2ac) while *E. faecalis* CCE induced differential expression of 14 *E. faecalis* specific genes, three *E. faecalis* and *S. aureus* common genes, and three *S. aureus*-specific genes (Figure 2abc). Of particular interest, *E. faecalis* CCE induced differential expression of the *S. aureus* biomarker *fmo-2*, a flavin-containing monooxygenase with a presumptive function of detoxification (Irazoqui et al. 2008). In summary, *E. faecalis* CCE and *E. faecalis* SE may indeed have evolved to stimulate pathogen-specific transcriptional responses, with *E. faecalis* CCE inducing more specific DEGs than *E. faecalis* SE.

I next investigated if evolved *E. faecalis* induced differential expression of *clec* genes since CTLDs are key in responding to microbe-associated molecular patterns and can even be microbe-specific (Pees et al. 2016). I summarized expression changes for *clec* DEGs from *E. faecalis* CCE and *E. faecalis* SE exposures (Table 2). The *E. faecalis* CCE treatment induced differential expression of nine *clec* genes and the *E. faecalis* SE treatment three *clec* genes. The only shared expression change was downregulation of *clec-48*, which is localized in the intestine (Mallo et al. 2002). The *E. faecalis* SE treatment significantly downregulated clec-48, 49, and 50, which are genetic paralogues that encode homologous proteins (Howe et al. 2016; Ortiz et al. 2014; Spencer et al. 2011), and all of which are also DEGs upon *E. faecalis* Anc exposure. The *E. faecalis* CCE treatment, on the other hand, influenced differential expression of diverse *clec* genes, only 4/9 of which are DEGs with *E.*

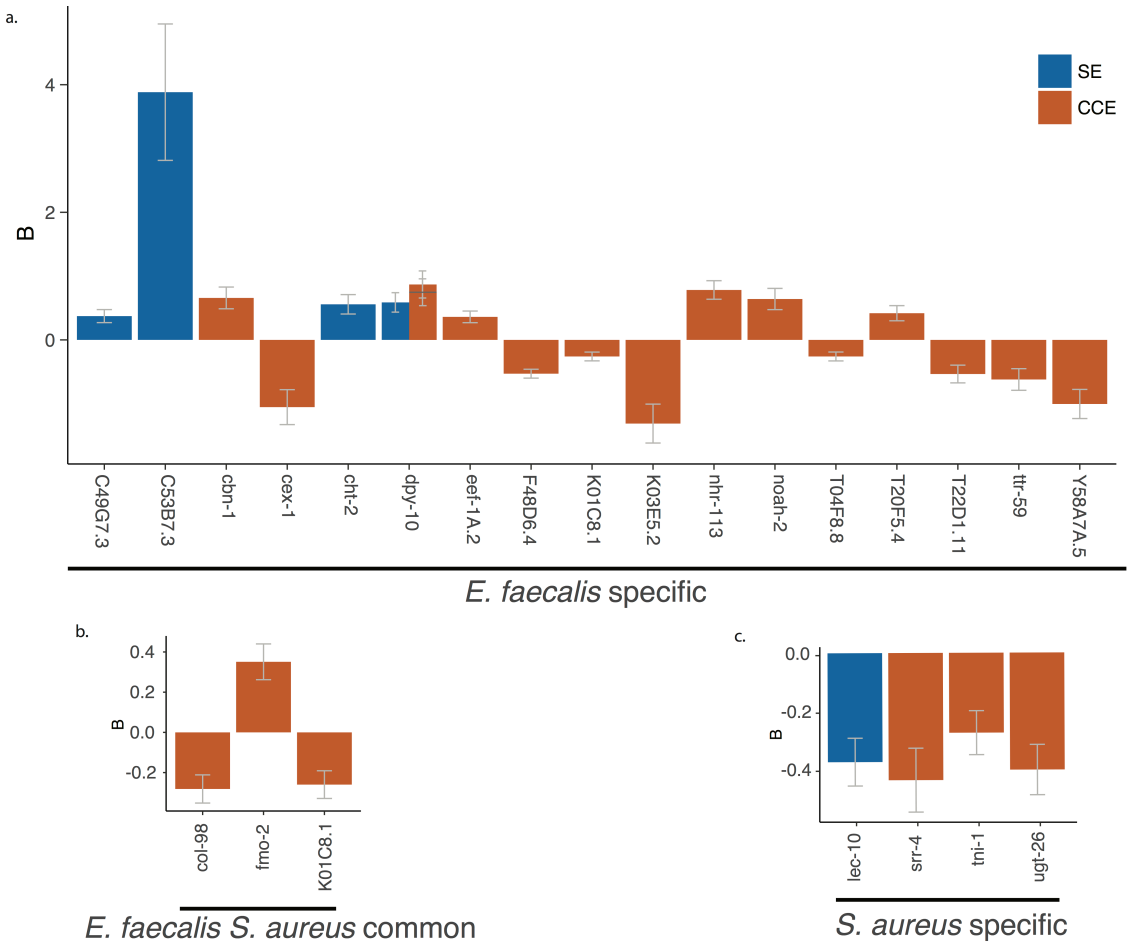


Figure 2. Pathogen specific and common genes significantly differentially expressed in *C. elegans* under *E. faecalis* SE and *E. faecalis* CCE exposures. Significant DEGs from *C. elegans E. faecalis* SE and *E. faecalis* CCE exposures identified as a. *E. faecalis* specific, b. *E. faecalis* and *S. aureus* common or c. *S. aureus* specific. Y-axis shows β-values and x-axis gene names. Multiple transcripts within DEG are denoted by split bars (e.g., dpy-10). Error bars ± s.e.

Table 2 clec gene β-values.

clec -	CCE/AE	SE/AE	*E. faecalis*/ OP50	Evidence
48	-0.39	-0.42	2.1	10.1016/j.ydbio.2006.10.024
49		-0.41	0.49	10.1242/dev.02185
50		-0.3	0.18	10.1242/dev.02185
136	-2.37			10.1242/dev.00914
137	-3.53		0.67	10.1016/j.ydbio.2005.05.017
138	-3.13		1.74	10.1016/j.ydbio.2005.05.017
146	-0.35		1.001	10.1101/gr.114595.110
180	0.78			10.1534/g3.115.022517 10.1016/j.ydbio.2010.05.502
197	-2.4			10.1016/j.ydbio.2005.05.017
208	-3.4			-
219	-3.36			10.1016/j.celrep.2016.09.051

clec genes that were significantly differentially expressed in E. faecalis CCE exposure compared to *E. faecalis* Anc exposure, and *E. faecalis* SE compared to *E. faecalis* Anc exposure (Wald-test; adj-P<0.05). β-value also shown if *clec* gene was significant with *E. faecalis* exposure. *E. faecalis* and *E. faecalis* Anc are the same.

faecalis Anc exposure. Again, this suggests that *E. faecalis* CCE induced differential expression of more microbe-associated genes than *E. faecalis* SE.

GO terms functionally enriched with E. faecalis SE and E. faecalis CCE exposures

To explore functional roles that exposures to *E. faecalis* SE and *E. faecalis* CCE might regulate in *C. elegans* compared to *E. faecalis* Anc, I investigated which GO terms were significantly enriched with treatments' DEGs (Figure 3ab). With exposure to *E. faecalis* SE, four GO terms were significantly functionally enriched (Figure 3b), where GO:0042329 (structural constituent of collagen and cuticulin-based cuticle) showed the highest fold enrichment (Figure 3ab). With exposure to *E. faecalis* CCE, 21 GO terms were significantly functionally enriched with DEGs (Figure 3c), 3/21 of which overlapped with GO terms from the SE treatment (GO:0042302; GO:0005581; GO:0005576). Four GO terms from the CCE treatment were related to oxidoreductase activity, oxidation-reduction processes or monooxygenase activity, and the most downregulated oxidation-related DEG by CCE was *skn-1*, a pathogen-related redox regulator (Papp et al. 2012; van der Hoeven et al. 2011). *E. faecalis* CCE also functionally enriched epithelial development (GO:0002054), and each of the genes mapping to this term were upregulated. For both treatments, I provided a complete list of significantly enriched GO terms and their associated genes (Supplementary Table 3). In comparison to the *E. faecalis* Anc treatment, these results suggest that *E. faecalis* SE exclusively alters collagen and cuticle-related transcriptional responses while *E. faecalis* CCE also induces these responses but amongst a vaster functional response including functions related to oxidation, epithelial development, and heme and iron binding.

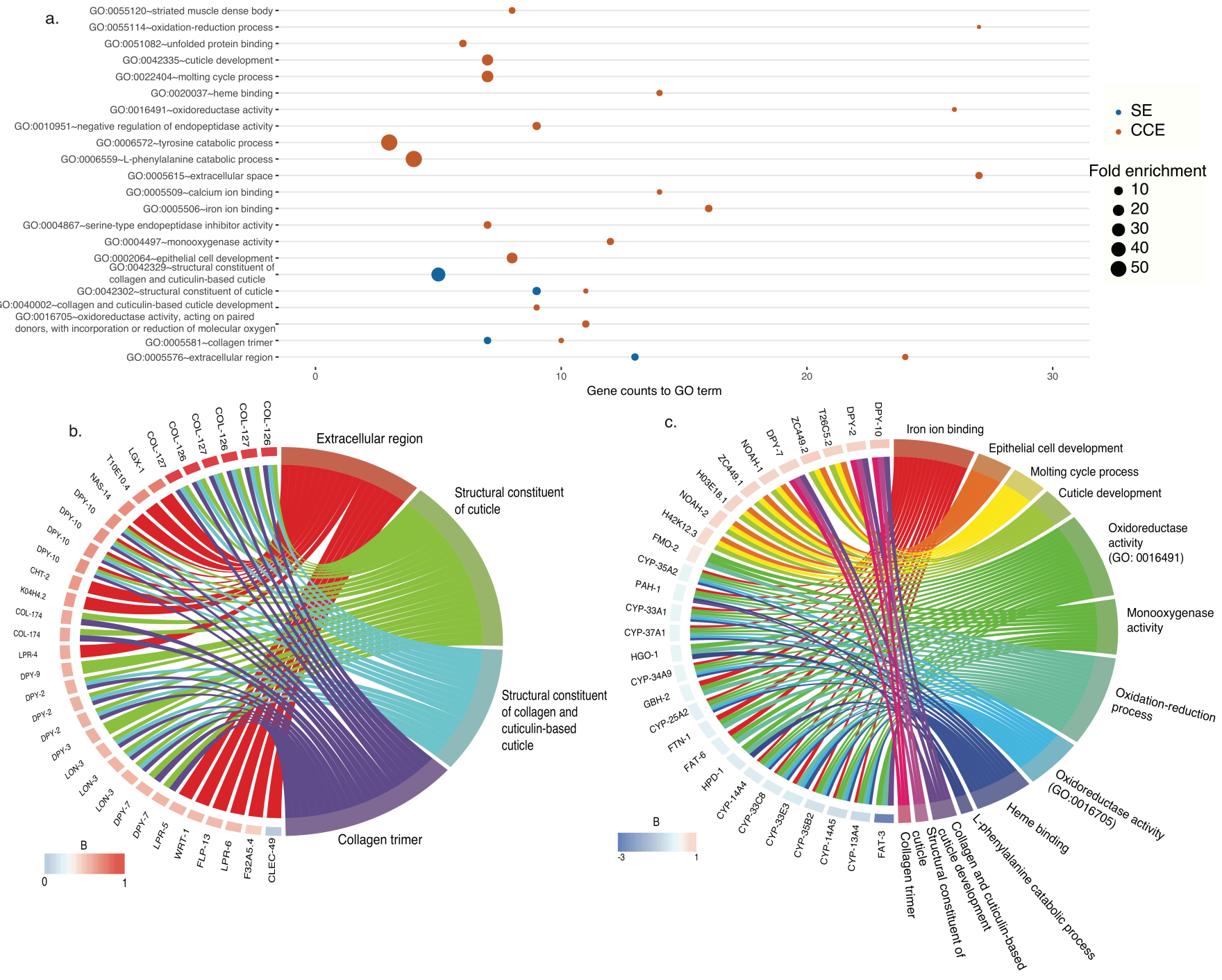

a.
GO:0055120~striated muscle dense body
GO:0055114~oxidation-reduction process
GO:0051082~unfolded protein binding
GO:0042335~cuticle development
GO:0022404~molting cycle process
GO:0020037~heme binding
GO:0016491~oxidoreductase activity
GO:0010951~negative regulation of endopeptidase activity
GO:0006572~tyrosine catabolic process
GO:0006559~L-phenylalanine catabolic process
GO:0005615~extracellular space
GO:0005509~calcium ion binding
GO:0005506~iron ion binding
GO:0004867~serine-type endopeptidase inhibitor activity
GO:0004497~monooxygenase activity
GO:0002064~epithelial cell development
GO:0042329~structural constituent of collagen and cuticulin-based cuticle
GO:0042302~structural constituent of cuticle
GO:0040002~collagen and cuticulin-based cuticle development
GO:0016705~oxidoreductase activity, acting on paired donors, with incorporation or reduction of molecular oxygen
GO:0005581~collagen trimer
GO:0005576~extracellular region
0
10
20
30
Gene counts to GO term
SE
CCE
Fold enrichment
10
20
30
40
50
b.
Extracellular region
Structural constituent of cuticle
Structural constituent of collagen and cuticulin-based cuticle
Collagen trimer
COL-126
COL-127
COL-126
COL-127
COL-126
COL-127
COL-126
LGX-1
T10E10.4
NAS-14
DPY-10
DPY-10
DPY-10
DPY-10
DPY-10
CHT-2
K04H4.2
COL-174
COL-174
LPR-4
DPY-9
DPY-2
DPY-2
DPY-2
DPY-3
LON-3
LON-3
LON-3
DPY-7
DPY-7
LPR-5
WRT-1
FLP-13
LPR-6
F32A5.4
CLEC-49
B
0
1
c.
Iron ion binding
Epithelial cell development
Molting cycle process
Cuticle development
Oxidoreductase activity (GO: 0016491)
Monooxygenase activity
Oxidation-reduction process
Oxidoreductase activity (GO:0016705)
Heme binding
L-phenylalanine catabolic process
Collagen and cuticulin-based cuticle development
Structural constituent of cuticle
Collagen trimer
DPY-10
DPY-2
T26C5.2
ZC449.2
DPY-7
NOAH-1
ZC449.1
H03E18.1
NOAH-2
H42K12.3
FMO-2
CYP-35A2
PAH-1
CYP-33A1
CYP-37A1
HGO-1
CYP-34A9
GBH-2
CYP-25A2
FTN-1
FAT-6
HPD-1
CYP-14A4
CYP-33C8
CYP-33E3
CYP-35B2
CYP-14A5
CYP-13A4
FAT-3
B
-3
1

Figure 3. GO term analysis of significant DEGs from comparing *C. elegans* under *E. faecalis* SE and *E. faecalis* CCE exposures to *E. faecalis* Anc exposure. GO terms significantly enriched with significant DEGs from *C. elegans* exposed to *E. faecalis* CCE and *E. faecalis* SE compared to *E. faecalis* Anc exposure. Significant DEGs from sleuth (Wald- Test; adj-P < 0.05) were investigated for functional enrichment using DAVID 6.8 (2016 build). **a.** Counts of DEGs mapped to significantly enriched GO terms and GO term fold enrichment (adj-P < 0.05). **b.** Chord plot of significantly enriched GO terms of *C. elegans* exposed to *E. faecalis* SE compared to *E. faecalis* Anc with mapping of DEGs to GO terms (adj-P < 0.05). **c.** Chord plot of significantly enriched GO terms of *C. elegans* exposed to CCE compared to *E. faecalis* Anc with mapping of DEGs to GO terms; dataset pruned to where GO terms must map to at least three genes and at least three genes must be assigned to a term (adj-P < 0.05). b. and c. heatmaps show β-values from Wald-Test. N = 4 biological replicates per treatment.

To more directly investigate how the evolution of a symbiont in the presence of a pathogen can induce transcriptional responses different to a symbiont absent a pathogen, I directly compared DEGs and GO terms from the *E. faecalis* CCE treatment to the *E. faecalis* SE treatment. Methodologically, this meant comparing the RNA profiles of the four *C. elegans* populations exposed to *E. faecalis* CCE to the populations of *C. elegans* exposed to *E. faecalis* SE. This revealed 84 significant DEGs, with an average absolute β-value of 0.96 ± s.e. 0.14 (Supplementary Table 4). Mapping these DEGs to GO terms, I found that 14 DEGs were associated with the 10-fold enriched GO term innate immune response (GO:0045087) (Figure 4; DAVID 6.8 (2016 build); P-adj < 0.01). In fact, this was the only significantly enriched GO term from DEGs comparing *E. faecalis* CCE to *E. faecalis* SE. The next GO term to nearest to marginal significance was defense response (GO:0006952) (Figure 4; DAVID 6.8 (2016 build); P-adj = 0.073), which is an ancestor GO term to innate immune response (Supplementary Figure 3; EMBL-EBI QuickGO). Though the innate immunity GO term also appeared when comparing *E. faecalis* CCE to *E. faecalis* Anc, it was not significant since more enriched GO terms enriched overall likely resulted in more stringent adjusted p-values. In short, these results reveal that the only functional enrichment difference between *C. elegans E. faecalis* CCE and *E. faecalis* SE is innate immune regulation.

I also provide a table of the DEGs mapping to the significantly enriched GO terms from the *E. faecalis* CCE to *E. faecalis* SE comparison with description and references (Table 3). I highlight whether these DEGs are also differentially expressed upon *E. faecalis* and *S. aureus* (Irazoqui et al. 2007) exposures (Table 3). Interestingly, with the exception of *fmo-2*, the *S. aureus* upregulated DEGs also

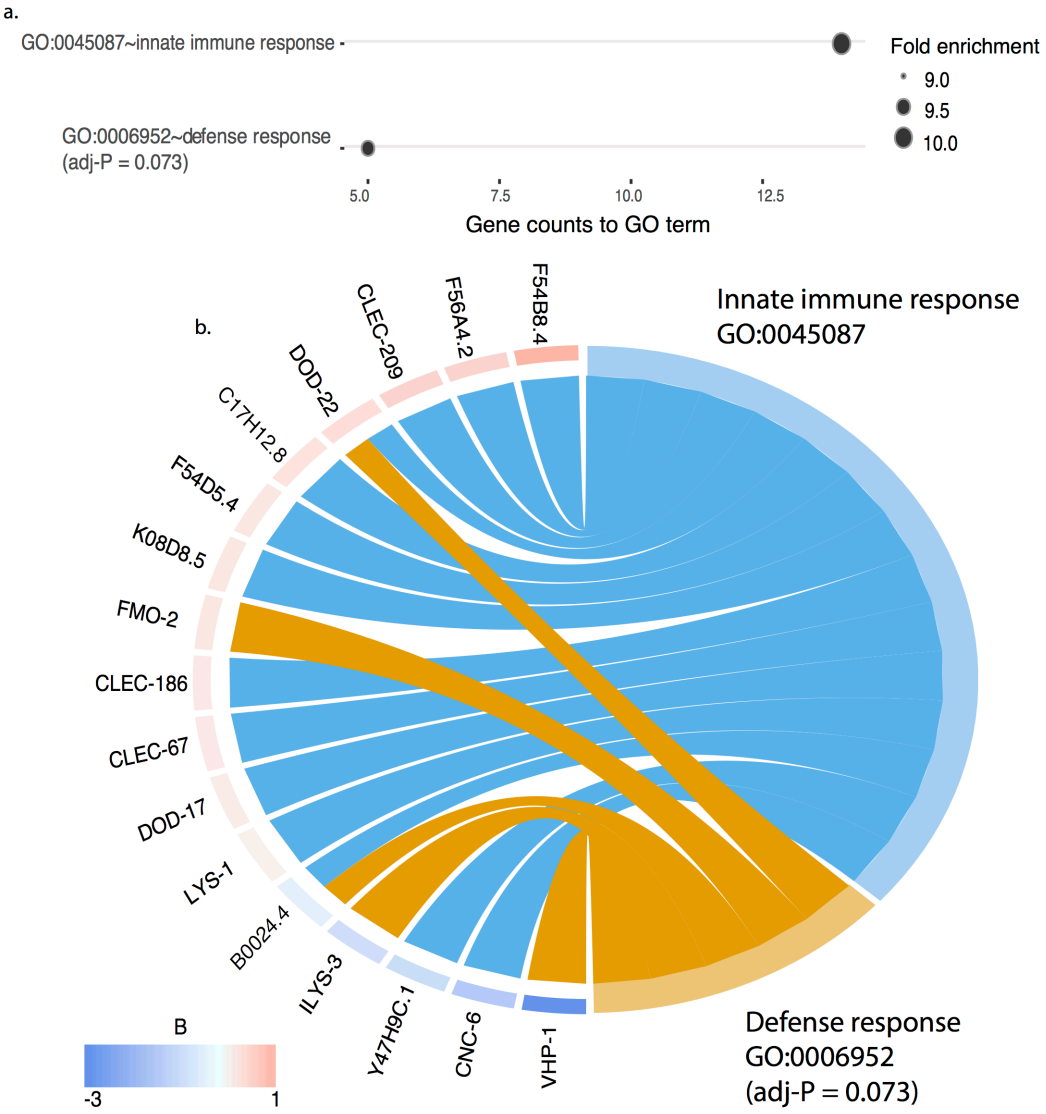


Figure 4. GO term analysis of significant DEGs from comparing *C. elegans* under *E. faecalis* CCE exposure to *E. faecalis* SE exposure. Significantly enriched GO terms with significant DEGs from *C. elegans* exposed to *E. faecalis* CCE compared to *E. faecalis* SE exposure. Significant DEGs from sleuth (Wald- Test; adj-P < 0.05) were investigated for functional enrichment using DAVID 6.8 (2016 build). **a.** Counts of DEGs mapped to significantly enriched GO terms and GO term fold enrichment (adj-P < 0.05). **b.** Chord plot of significantly enriched GO terms of *C. elegans* exposed to *E. faecalis* CCE compared to *E. faecalis* SE with mapping of DEGs to GO terms (adj-P < 0.05). Heatmap shows β-values from Wald-Test.

Table 3. DEGs from *E. faecalis* CCE to *E. faecalis* SE comparison mapping to enriched GO terms.

Gene	CCE/ Anc	CCE/ OP50	S. aureus/ OP50	GO term	Description
B0024.4	-	+		Innate immune response, Defense response	Uncharacterized protein involved in defense response
C17H12.8				Innate immune response	
CLEC-186				Innate immune response	C-type lectin
CLEC-209		-		Innate immune response	C-type lectin
CLEC-67		+		Innate immune response	C-type lectin
CNC-6	-	+		Innate immune response	CaeNaCin (Caenorhabditis bacteriocin)
DOD-17			-	Innate immune response	Downstream Of DAF-16 (regulated by DAF-16)
DOD-22		+	+	Innate immune response, Defense response	Downstream Of DAF-16 (regulated by DAF-16)
F54B8.4		+		Innate immune response	Homolog of DAP-1, involved in apoptosis
F54D5.4		-		Innate immune response	
F56A4.2		-		Innate immune response	C-type lectin
FMO-2	+	+	+	Defense response	Dimethylaniline monooxygenase
ILYS-3	-	+	+	Defense response	Invertebrate lysozyme
K08D8.5		+		Innate immune response	
LYS-1			+	Innate immune response	Lysozyme
VHP-1		±		Defense response	Tyrosine-protein phosphatase vhp-1
Y47H9C.1	-		+	Innate immune response	

DEGs from Figure 4 and GO terms. Showing direction of change in other exposure comparisons (*E. faecalis* CCE/*E. faecalis* Anc, *E. faecalis* CCE/OP50 and S. aureus/*E, coli* OP50) (Alper et al. 2007; Irazoqui et al. 2010).

upregulated by *E. faecalis* (ilys-3, cnc-6, B0024.4, and Y47H9C.1) decreased in expression with *E. faecalis* CCE exposures and the DEGs also upregulated by *S. aureus* increased in expression in the *E. faecalis* CCE to *E. coli OP50* comparison (*fmo-2, ilys-3, dod-22*) (Table 3). For clarity, *ilys-3* significantly decreased relative to *E. faecalis* Anc exposure but increased relative to *E. coli* OP50 exposure.

Processing of 16S rRNA reads from C. elegans' natural microbiota

I next sought to investigate possible consequences of symbiont exposure, and specifically defensive mutualist *E. faecalis* CCE exposure, on shaping hosts' natural microbiota. To do so, I investigated the microbiome of treatment exposed *C. elegans* after rearing in microbial enriched compost environments. These compost environments are established as sufficient to maintain *C. elegans,* their microbiome constituents, and interactions between the two (Berg et al. 2016). For exposures, I used *E. faecalis* Anc; *E. faecalis* SE; *E. faecalis* CCE; a non-protective and non-colonizing control, *E. coli* OP50; and a naturally-isolated *C. elegans* protective microbe, *P. mendocina* (Montalvo-Katz et al. 2013). After initial microbial exposure, *C. elegans* were reared in compost for 24h then harvested and externally washed, after which their gut microbiomes were extracted and sequenced. Sequencing of the 16S rRNA V4 region on 75 *C. elegans* microbiome samples returned on average 46,317 reads at an average length of 253bp after quality filtering, de-replicating, cleaning sequences of chimeras, and removing sequences observed in an extraction control and non-template PCR controls. After further preprocessing (Supplementary file 4), I retained 65 samples with an average of 50,903 reads per sample and an average of 64 ribosomal sequence variants (RSVs)

Table 4. Alpha diversity measurements of *C. elegans* microbiota after compost exposure.

Treatment	*Observed RSVs*	*Shannon*	*Chao 1*
Anc	39.7 ± 5.00	1.47 ± 0.09	41.4 ± 5.57
SE	44.4 ± 4.56	1.40 ± 0.09	46.2 ± 4.84
CCE	50.0 ± 4.31	1.42 ± 0.05	51.5 ± 4.47
OP50	48.4 ± 8.99	1.55 ± 0.12	49.3 ± 8.93
Pm	28.4 ± 3.00	1.30 ± 0.12	28.6 ± 3.06

Treatments are of different exposures, prior to compost exposure. Observed RSV measurement ($F_{(4,39)} = 4.19$, $P < 0.01$). Shannon diversity measurements ($F_{(4,39)} = 0.478$, $P = 0.75$). Chao 1 diversity measurement ($F_{(4,39)} = 5.22$, $P< 0.05$). Showing means ± s.e.. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved.

per sample. Each RSV represents a unique microbial strain, as defined by the 16S sequence.

Effects of pre-exposure on microbiota diversity

I described exposure effects on microbiota diversity using both within (alpha) and between (beta) sample diversity measurements. For alpha diversity, I report mean and standard error measurements for observed RSVs, Shannon and Chao 1 diversity metrics (Table 3). Observed RSVs indicates the number of RSVs per sample, the Shannon metric is an equal weighted metric for species richness and evenness, and the Chao 1 index is a metric weighted towards rare RSVs that also incorporates richness and evenness. Treatment was a significant factor when modeling its effect on observed RSVs and Chao 1 diversity but not Shannon diversity (Figure 5abc; Supplementary tables 5-6), likely indicating major differences were driven by RSV richness and the abundance of rare RSVs. Further, post-hoc analyses revealed significant differences were driven by low RSV diversity in samples exposed to *P. mendocina*, where there were 1.77x significantly fewer RSVs in *C. elegans* exposed to *P. mendocina* compared to *E. faecalis* CCE and *E. coli* OP50 pre-exposures (ANOVA; Tukey-HSD; adj-P < 0.05; Supplementary table 5). Similarly there was on average of 1.8x lower Chao 1 diversity in *C. elegans* exposed to *P. mendocina* compared to *C. elegans* exposed to *E. faecalis* CCE (ANOVA; Tukey-HSD; adj-P < 0.05; Supplementary table 6). These results indicate that *E. faecalis* exposures had no significant effects on alpha diversity.

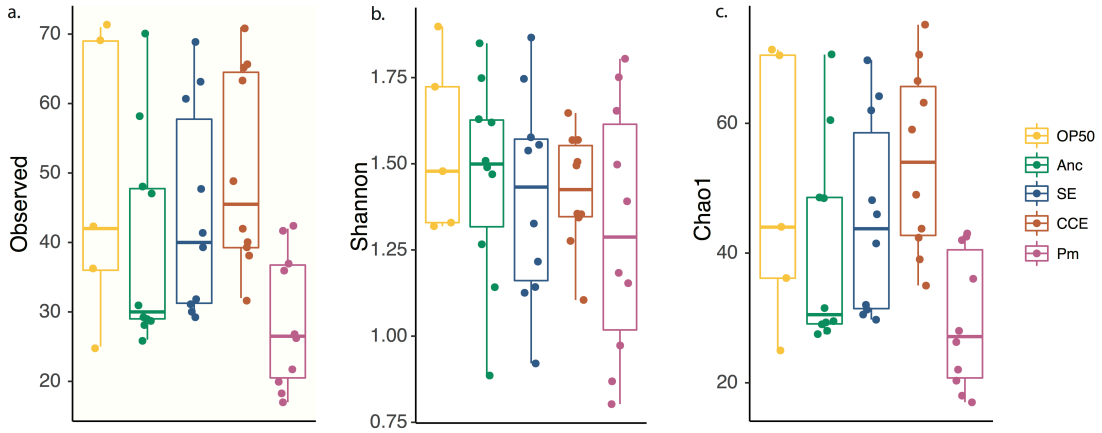


Figure 5. Alpha diversity measurements of *C. elegans* microbiota after compost exposure. Treatments are of different exposures, prior to compost exposure. **a.** Observed ribosomal sequence variant (RSV) measurement ($F_{(4,39)}$ = 4.19, P < 0.01). **b.** Shannon diversity measurements ($F_{(4,39)}$ = 0.478, P = 0.75). **c.** Chao 1 diveristy measurement ($F_{(4,39)}$ = 5.22, P< 0.05). Plotted with median (line), hinges as first and third quartiles (25th and 75th percentiles), and ends as ranges. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved. Pm = *P. mendocina.* OP50 = *E. coli* OP50.

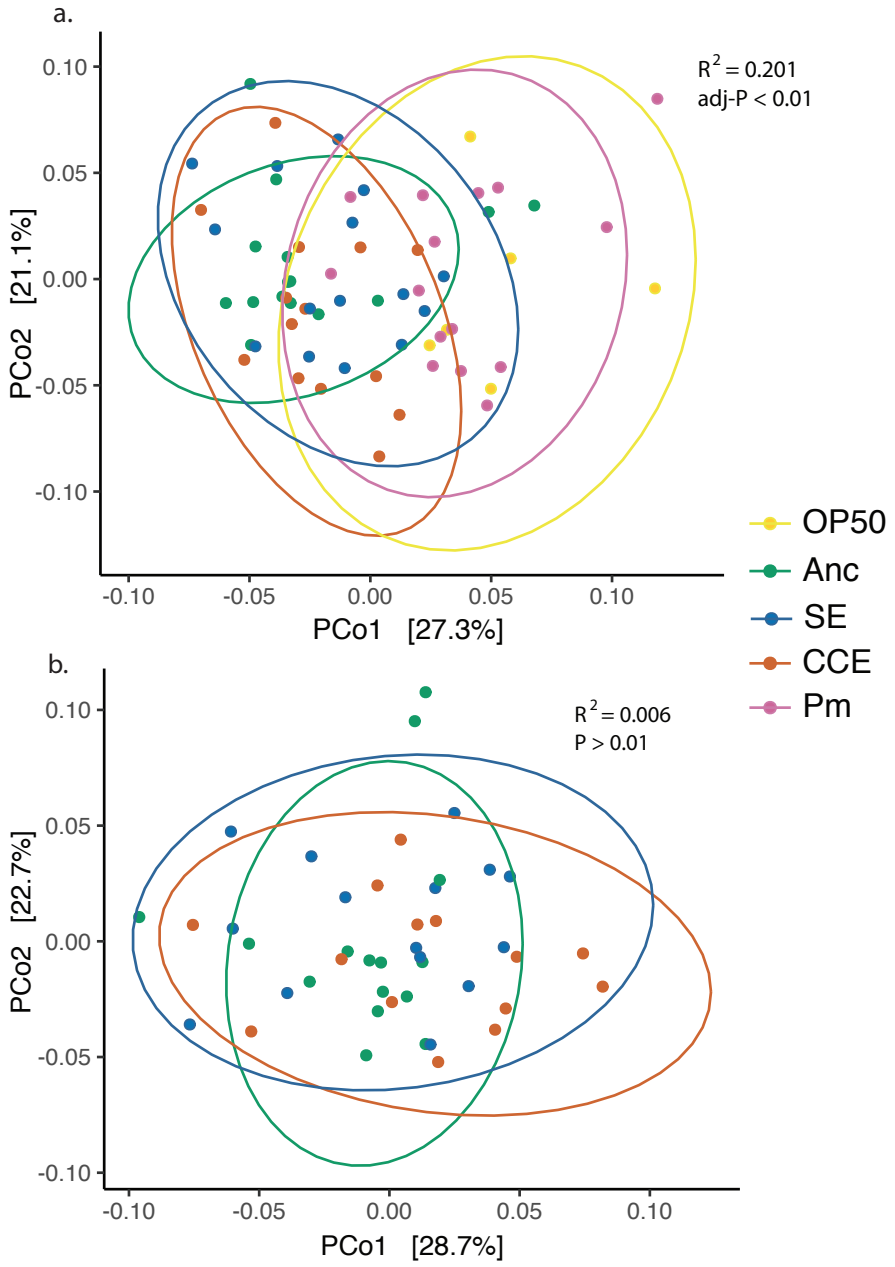


Figure 6. Principal coordinate analyses (PCoA) on weighted UniFrac scores of *C. elegans* microbiota. a. PCoA on weighted UniFrac scores by exposure treatment. Exposure treatment between all treatments worked as a significant predictor of ecosystem distance (ANOSIM; R^2 = 0.201; adj-P < 0.01; perm = 999). **b.** PcoA on weighted UniFrac scores comparing microbiota from *E. faecalis* strain exposures. Exposure treatment between *E. faecalis* strains did not work as a significant predictor of ecosystem distance (ANOSIM; R^2 = 0.201; adj-P = 0.340; perm = 999). Ellipses are drawn at 95% confidence intervals. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved. Pm = *P. mendocina.* OP50 = *E. coli* OP50.

In beta diversity analyses, the first two axes explained more than 50% of sample variance (Figure 6; PCo1 = 28.7% and PCo2 = 22.7%) and a marginal batch effect remained (ANOSIM; $R^2 = 0.083$; $P < 0.01$). Exposure treatment was a small but significant predictor of discernably clustering *C. elegans* microbiota diversity (Figure 6a; ANOSIM; $R^2 = 0.201$; $P < 0.01$), meaning treatments were more similar to one another than each other. However, when subset to only *E. faecalis* exposures (Anc, SE, and CCE), treatment was no longer a significant predictor of clustering (Figure 6b; ANOSIM; R^2; $P = 0.34$). Overall, this suggests that the observed small differences were primarily driven by differences between *E. faecalis* exposures as a species and not by *E. faecalis* strains.

Differentially abundant microbiota influenced by pre-exposure treatments

I also measured how treatments influenced differential abundance of microbiota members at the genus level. First, comparing *E. faecalis* (Anc, SE and CCE) and *P. mendocina* exposures to *E. coli* OP50, I observed that all three *E. faecalis* strains significantly increased the abundance of a RSV identified as *Enterococcus* (sq10; base mean = 1277), by an average of 12.4 log2fold (s.e. = 0.279) (Figure 7a). Interestingly, *P. mendocina* also increased *Enterococcus* abundance but by 6.08 log2fold (Figure 7a). *Enterococcus* was most abundant in *C. elegans* microbiota from *E. faecalis* exposures (mean relative abundance = 0.0190; s.e. 0.0045) and not found in *C. elegans* microbiota from the *E. coli* OP50 exposures (Figure 7b). I also found that *P. mendocina* and *E. faecalis* SE exposures significantly decreased abundance of a RSV previously identified core *C. elegans* microbiota genus,

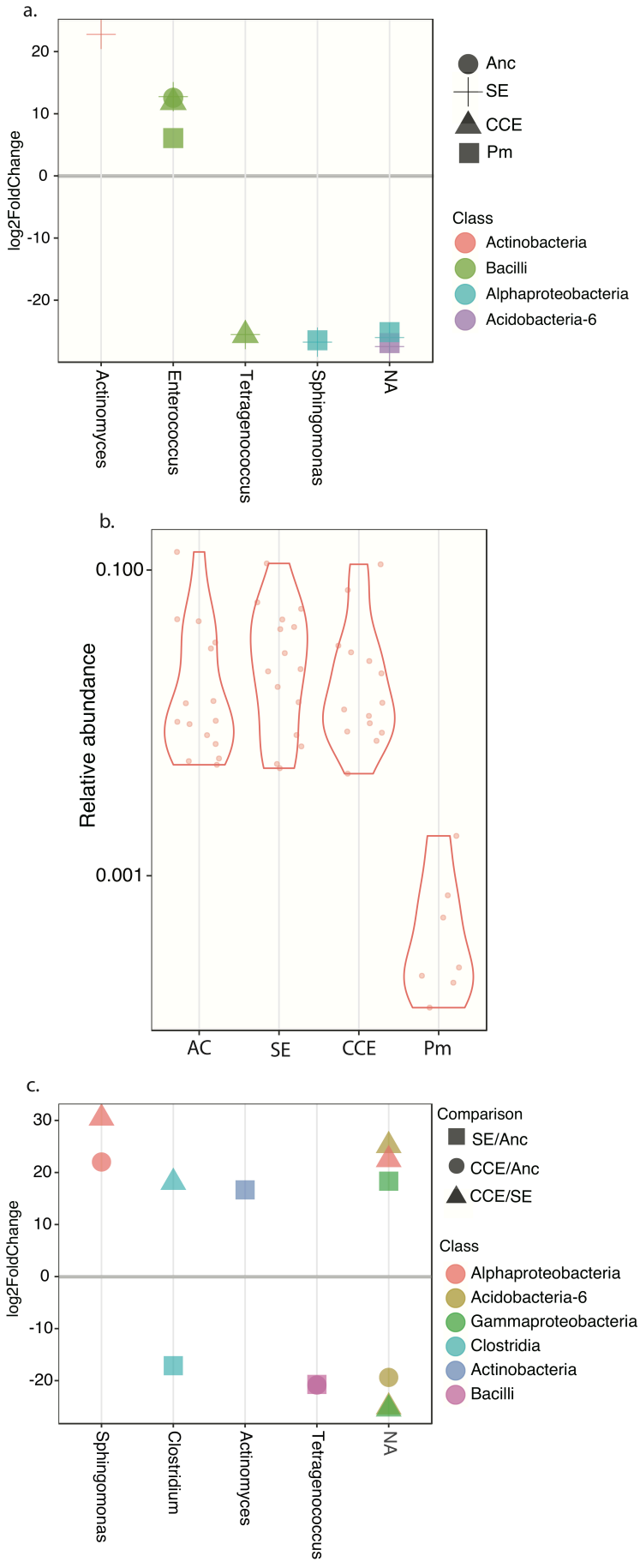

a.
log2FoldChange
20
10
0
-10
-20
Actinomyces
Enterococcus
Tetragenococcus
Sphingomonas
NA
Anc
SE
CCE
Pm
Class
Actinobacteria
Bacilli
Alphaproteobacteria
Acidobacteria-6
b.
Relative abundance
0.100
0.001
AC
SE
CCE
Pm
c.
log2FoldChange
30
20
10
0
-10
-20
Sphingomonas
Clostridium
Actinomyces
Tetragenococcus
NA
Comparison
SE/Anc
CCE/Anc
CCE/SE
Class
Alphaproteobacteria
Acidobacteria-6
Gammaproteobacteria
Clostridia
Actinobacteria
Bacilli

Figure 7. RSVs that significantly differ in abundance in *C. elegans* microbiota after different pre-exposure treatments and compost exposure. **a.** Log2fold change of significantly differentially abundant RSVs identified comparing microbiota of *C. elegans* exposed to different treatments (*E. faecalis* Anc, SE and CCE, and *P. mendocina*) over control (*E. coli* OP50) exposure (DESeq2; adj-P < 0.05). **b.** Violin plot of relative abundance of *Enterococcus*, sq10, in *C. elegans* microbiota after exposure treatments and compost exposure. *Enterococcus* was not observed in microbiota of *C. elegans* exposed to *E. coli* OP50. **c.** Log2fold change of significantly differentially abundant RSVs identified comparing microbiota of *C. elegans* exposed to *E. faecalis* CCE to *E. faecalis* SE, and *E. faecalis* CCE and *E. faecalis* SE to *E. faecalis* Anc. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved. Pm = *P. mendocina.*

Sphingomonas (Dirksen et al. 2016) (sq256; base mean = 0.481), by an average of 26.6 log2fold (s.e. = 0.149).

I also measured how *E. faecalis* strain exposures influenced differential abundance of microbiota between (Figure 7c). Compared to *E. faecalis* Anc, *E. faecalis* CCE exposure led to differential abundance of three RSVs and *E. faecalis* SE of four RSVs, with the only shared one being *Tetragenococcus* (sq103; base mean = 2.92). This genus similarly decreased in abundance after both exposures (Figure 7c). Interestingly, compared to both *E. faecalis* SE and *E. faecalis* Anc, *E. faecalis* CCE significantly influenced an increase of the aforementioned core microbe, *Sphingomonas,* by an average of 26.2 log2fold (s.e. 4.21). The two sequences with identified genera (sq103, *Tetragenococcus*; sq265, *Clostridium*) that *E. faecalis* CCE significantly decreased in abundance compared to *E. faecalis* Anc are Gram-positive. In addition, neither of the genera found in the compost samples that can be *C. elegans* pathogens (*Bacillus* and *Pseudomonas*) (Griffitts et al. 2003; Wareham et al. 2005) increased in abundance with pre-exposure to evolved *E. faecalis* strains. For all differential abundance comparisons I supply supplementary tables with log2fold changes, RSV base means, adj-P values and deepest available taxonomic classifications (Supplementary table 7).

C. elegans transcript correlations with Enterococcus abundance

To see if *E. faecalis*-related transcripts that decreased in *C. elegans* upon evolved *E. faecalis* exposure related to increased accumulation of *Enterococcus* in compost exposures I tested for correlations between transcript abundance prior to compost and *Enterococcus* relative abundance in *C. elegans* exposed to compost. As

candidates I used *clec-48*, the only *clec* DEG that decreased in abundance when comparing both *E. faecalis* SE and *E. faecalis* CCE exposures, and *ilys-3*, a DEG that decreased in abundance with *E. faecalis* CCE exposure and is related to the defense response GO term (Figure 4). My results indicate that decreased expression of either of these transcripts worked as predictors of *Enterococcus* relative abundance after compost exposure (Pearson's; Ps > 0.05). Species or strain level sequence classification for *Enterococcus* with 16S sequences was not available.

Evolved E. faecalis colonization efficacy and protection persistence

To investigate phenotypic outcomes proposed to arise from *in vivo* symbiont evolution (Hoang et al. 2016), I assayed how evolution of *E. faecalis* CCE resulted in increased colonization efficacy and protection persistence. Upon initial *E. faecalis* exposures, *C. elegans* were colonized by, on average, 3.43x more *E. faecalis* CCE colony forming units (CFUs) (mean = 8201 CFUs; s.e. = 1540), than *E. faecalis* Anc (mean = 2664; s.e. = 543) and *E. faecalis* SE (mean = 2125; s.e. = 365), a finding that was significant (Figure 8a; T-test; adj-P < 0.05).

Next, since protective effects by symbionts persist but can be diluted in natural contexts (Siven et al. 2015; Lenhart & White 2017), I investigated protection persistence after exposure to natural microbial contexts. I found that protection by *E. faecalis* is maintained amongst a natural microbiome, where mortality upon direct *Staphylococcus aureus* exposure after compost exposure was 72.7% when exposed to *E. faecalis* CCE, a 23.1% lower mortality than the other *E. faecalis* exposures (Figure 8b; Wilcoxon test; adj-P < 0.05).

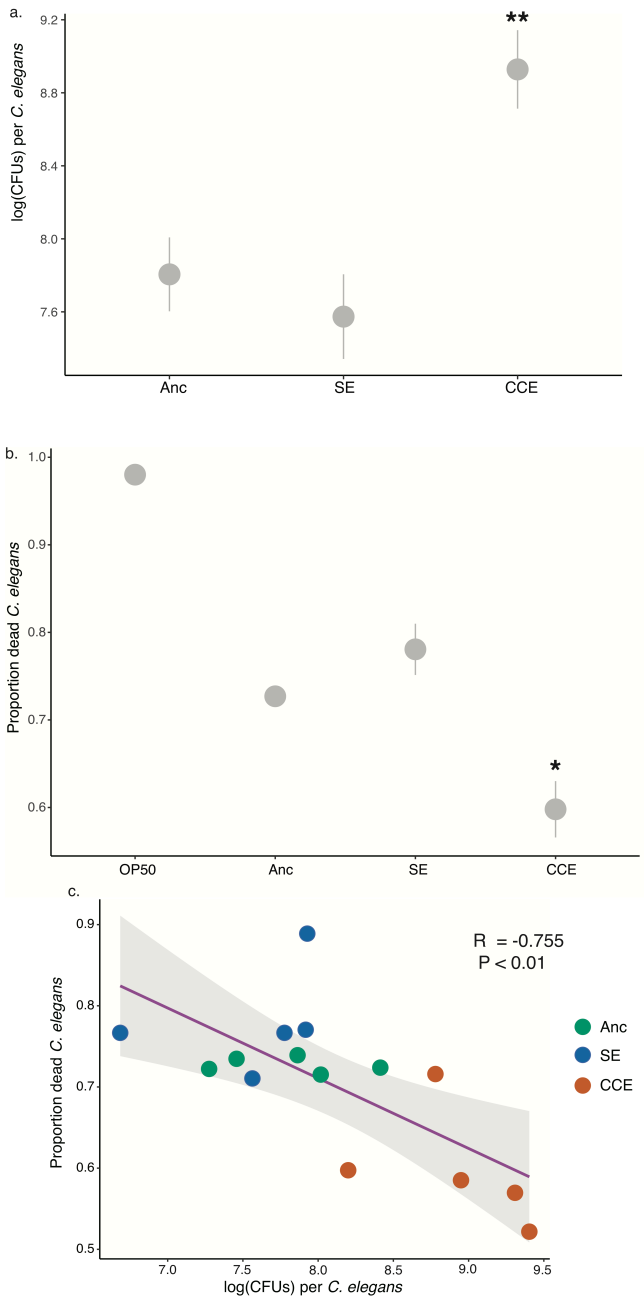


Figure 8. Evolved *E. faecalis* strains colonization in *C. elegans* and effects on *S. aureus* induced mortality amongst natural microbiome. a. *C. elegans* gut bacterial CFUs after exposure to *E. faecalis* Anc, *E. faecalis* SE, or *E. faecalis* CCE (paired T-test; adj-P < 0.05). **b.** *C. elegans* mortality after different exposures and compost exposure and exposure to *S. aureus* (paired Wilcoxon test; adj-P < 0.05). **c.** Correlation between exposure gut colonization abundance and mortality under *S. aureus* infection after compost exposure (Pearson's; R = -0.775; T = -4.42; df = 13; P << 0.01). Data for CFUs and transcript levels collected at the same time points and from same batches, hence direct comparisons. n = 5 populations per treatment. Error bars = ± s.e. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved. Pm = *P. mendocina.* OP50 = *E. coli* OP50.

My results also indicate that initial colonization was a significant predictor of mortality after compost exposure, where increased *E. faecalis* accumulation prior to compost exposure resulted in decreased mortality upon *S. aureus* exposure after compost exposure (Figure 8c; R = -0.775; T = -4.42; df = 13; p << 0.01). Though I also examined whether *E. faecalis* colonization predicted relative abundance of *Enterococcus* post compost exposure, it did not (Pearson's; P > 0.05; Supplementary figure 4). In addition, relative abundance of *Enterococcus* post compost exposure did not predict decreased *S. aureus* induced mortality (Pearson's; P > 0.05; Supplementary figure 5).

C. elegans transcript correlations with E. faecalis colonization efficacy

I hypothesized that downregulation of *E. faecalis*-related transcripts would be linked with increased colonization and thus tested correlations between *E. faecalis* and downregulated immune-related transcripts. I observed that decrease expression of one candidate, *clec-48*, was not a significant predictor of colonization (Pearson's; P > 0.05; Supplementary figure 6), but decreased expression of *ilys-3* was in fact a very strong predictor of colonization (Figure 9; Pearson's; R = -0.999; P < 0.05). Other downregulated transcripts that similarly mapped to innate immunity and defense response GO terms did not predict colonization (Figure 4) (Pearson's; Ps > 0.05; Supplementary figure 7).

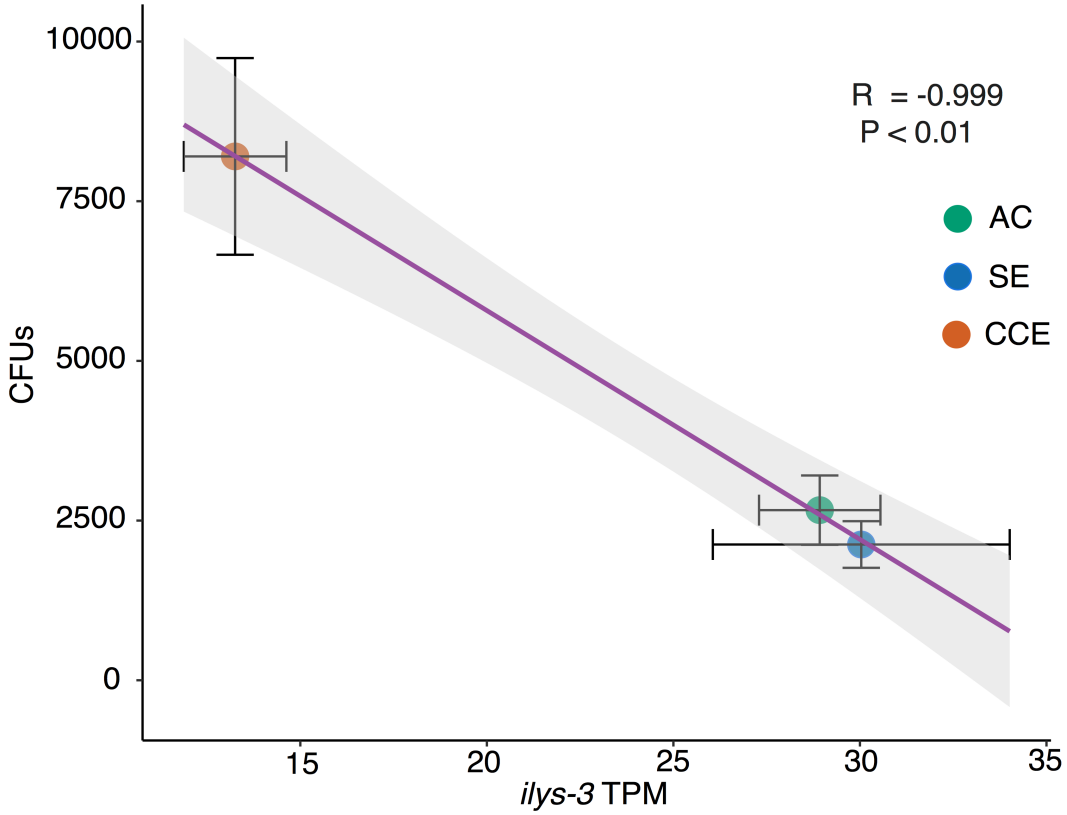


Figure 9. Correlation between *E. faecalis* CFUs in *C. elegans* guts and *ilys-3* TPM values. *C. elegans* gut bacterial CFUs after exposure to *E. faecalis* Anc, *E. faecalis* SE, or *E. faecalis* CCE correlated with transcript per million (TPM) values for transcripts identified as *ilys-3* from RNASeq experiments. Decreased *ilys-3* abundance is a significant predictor of *E. faecalis* CFUs, where the most CFUs and fewest transcripts are observed with *E. faecalis* CCE colonization (Pearson's; R = -0.999; P < 0.01). Data for CFUs and transcript levels collected at the same time points but from different batches and are means. CFUs collected from n = 5 replicate populations per treatment. RNASeq collected from n = 4 replicate populations per treatment. Error bars = ± s.e.. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved.

Discussion

Defensive microbes offer important protective benefits to host physiology in natural and applied settings (Oliver et al. 2013; Cosseau et al. 2008). They can offer protection through their influence on their hosts and hosts' microbiomes (Sorg & Sonenshein 2008; Doremus & Oliver 2017). Since ecological and evolutionary forces on hosts instigate effects on their microbiome and vice versa (Moeller et al. 2016; King et al. unpublished data), the host-microbiome system is inextricable. Thus, we hypothesized that *E. faecalis* CCE, a net mutualist evolved *in vivo* that protects against *S. aureus* infection, would, amongst protecting, affect its whole host-microbiome system. On the host end, we observed that *E. faecalis* CCE stimulated distinct transcriptional responses indicative of protection and colonization. And, that *E. faecalis* CCE colonized better than *E. faecalis* Anc or *E. faecalis* SE. Influencing the *C. elegans* microbiome, *E. faecalis* CCE had minimal impact overall. Additionally, *E. faecalis* CCE maintained protection against *S. aureus* even amongst the natural *C. elegans* microbiome. These results support previous findings that protective symbiont strains affect distinct host responses (K.-H. Lee & Ruby 2004), and describe novel ways in which symbionts affect microbiomes and maintain their phenotypic benefits amongst natural microbiota.

E. faecalis CCE effects on C. elegans

Previous findings show little to no similarity between independent studies DEGs from *C. elegans* microbial exposure (Doublet et al. 2017; Han et al. 2016; Wong et al. 2007; Mallo et al. 2002; Troemel et al. 2006; Shapira et al. 2006). This is despite using the same bacterial strains and similar culture conditions. The main

difference driving different transcriptional readouts could be age at time of RNA harvest (Boeck et al. 2016). For instance, upon *P. aeruginosa* PA14 exposure Troemel et al. (2006) harvested young adults and Shapira et al. (2006) harvested L4s and only revealed approximately 20% similarity in transcriptomes. Nonetheless, even though I did not harvest *C. elegans* exposed to *E. faecalis* at the same time point as the comparison study (Wong et al. 2007), I revealed a substantial amount of previously observed DEGs (>60%). My finding of high similarity is likely due to the use of RNASeq, while previous studies have used microarrays. Though comparing technologies is beyond the scope here, RNASeq's lack of probe bias and ability to reveal an altogether broader dynamic range of transcripts likely revealed more previously observed DEGs. Altogether, since these DEGs were revealed in different studies with different technologies, they should be considered robust markers of *E. faecalis* infection.

I revealed that *E. faecalis* CCE functionally enriched several GO terms related to oxidation-reduction processes. In addition, the gene most downregulated by *E. faecalis* CCE relative to *E. faecalis* Anc was *skn-1*, a gene involved in pathogen response and regulating homeostasis of host redox under infection (McCallum & Garsin 2016; Papp et al. 2012; van der Hoeven et al. 2011). *E. faecalis* CCE inhibits growth of *S. aureus in vitro* through the production of superoxides (King et al. 2016). Additionally, superoxides can play substantial roles in regulating innate epithelial immunity in the gut of *C. elegans* (McCallum & Garsin 2016; K.-A. Lee et al. 2015; Kim & W.-J. Lee 2014). Thus, it seems possible that *E. faecalis* CCE regulates the oxidation-reduction process in the *C. elegans* gut via production of its superoxides. Future experiments should assay superoxide production of *E. faecalis* CCE *in vivo*

and produce *E. faecalis* CCE strains that fail to produce superoxides and show that phenotypic or transcriptomic effects no longer subsist. In addition, one could assay superoxide importance by instigating superoxide production *in vivo* with redox-active heterocycles, such as paraquat, followed by *S. aureus* exposure. Experiments should also test the importance of *C. elegans* superoxide regulation by conducting protection assays using *C. elegans* with superoxide dismutase knockouts.

Common response genes to different pathogens are considered constituents of shared host responses to different infections (Wong et al. 2007). In this case, common response genes influenced by *E. faecalis* SE and *E. faecalis* CCE can be considered constituents of symbiont evolution, since both are *E. faecalis* and from lineages passaged in *C. elegans in vivo*. Shared DEGs were highly correlated in expression levels. And, shared DEGs and GO terms primarily related to *C. elegans* external collagen and cuticle expression. For example, several *dpy* genes (e.g., *dpy-10*) were shared. These genes encode the most external collagen and cuticle and are involved in general stress response mechanisms. (Taffoni & Pujol 2015; Wheeler & Thomas 2006). Since the surface of the cuticle is associated with pathogen immune evasion and pathogen adherence (Blaxter et al. 1992; Page et al. 1992), evolution of altered expression of associated genes may be related to active bacterial evasion. To test this hypothesis, I could use *C. elegans* with *dpy-10* knockouts and assay if *E. faecalis* CCE colonizes better and offers higher protection, and *E. faecalis* Anc and SE infect more effectively in this mutant.

Comparing both *E. faecalis* CCE and *E. faecalis* SE to *E. faecalis* Anc, *E. faecalis* CCE stimulated a substantially vaster transcriptional response. Further, directly comparing *E. faecalis* CCE to *E. faecalis* SE, the only significantly functionally

enriched GO term was innate immune response, and several DEGs also mapped to its ancestor GO term, defense response. Two particularly interesting innate immune response upregulated genes were *lys-1* and *fmo-2*. Previous extensive characterization of *lys*-1 shows that it is a key anti-microbial immune effector with common expression induced by pathogens including *S. marcescens, P. aeruginosa,* and *S. aureus* (Shapira et al. 2006; Alper et al. 2007; Irazoqui et al. 2010; Schulenburg et al. 2008). And, *fmo-2* a flavin-containing monooxygenase, is upregulated 100-fold upon *S. aureus* infection, making it the top ranking *S. aureus* biomarker (Irazoqui et al. 2010). These results may suggest upregulation of *lys-1* and *fmo-2* are related to *E. faecalis* CCE immune priming of *S. aureus* related genes. Such a mechanism would be similar to the one instigated by the defensive microbe, *P. mendocina,* in which it primes *P. aeruginosa*-related genes to prevent *P. aeruginosa* infection in *C. elegans* (Montalvo-Katz et al. 2013). It is also possible that these genes are related to strain-specific protection, which would explain phenotypic strain-specific protection by *E. faecalis* CCE (King et al. 2016). Protection assays with *E. faecalis* CCE and subsequent *S. aureus* exposure using *C. elegans* with knockouts at *fmo-2* or *lys-1*, or both, could reveal the importance of these genes for protection.

Strain-level specificity of symbionts and specificity of their protective mechanisms are observed in other systems. For instance, *Haminotella* strains protect clonal pea aphids from parasitoid wasps to varying degrees, ranging from 19% to nearly 100% (Oliver et al. 2005). Interestingly, the most protective *Haminotella* strain can protect across a range of aphid host genotypes (Oliver et al. 2005). Future research should expose diverse *Caenorhabditis* genera to *E. faecalis*

CCE to reveal the generality of *E. faecalis* CCE protection and the strength of its symbiont-by-nematode genotype interactions.

Amongst the innate immune response genes I also observed downregulation of an *E. faecalis* infection biomarker (Wong et al. 2007), *ilys-3*, an invertebrate lysozyme (Gravato-Nobre et al. 2016). Further, I found that downregulation of *ilys-3* was strongly correlated with increased *E. faecalis* colonization. Previous work shows that *ilys-3* expression in *C. elegans* is required for pharyngeal grinding, is expressed as an antibacterial effector in the intestine, and exhibits lytic activity against Gram-positive bacteria (Gravato-Nobre et al. 2016). Also, invertebrate lysozymes are common to numerous other organisms, including pea aphids (Gerardo et al. 2010) and mosquitoes (Paskewitz et al. 2008). Indeed, innate immune genes are often key components that underlie colonization and symbioses (Nyholm & Graf 2012).

E. faecalis CCE colonized better than both other *E. faecalis* strains. Symbionts can downregulate host responses to promote host-symbiont homeostasis (Park et al. 2016) and increase symbiont colonization (Cosseau et al. 2008). In fact, even strains can have different colonization efficacies (K.-H. Lee & Ruby 2004), a phenomenon that can even be explained by single gene level differences (Mandel et al. 2009). For instance, a single gene in *Vibrio fischeri* ES114 substantially promotes colonization in Hawaiian squid *Euprymna scolopes* (Mandel et al. 2009). Future work should investigate how *ilys-3* is linked with decreased lysozyme activity, increased colonization, and beneficial invertebrate-symbiosis interactions.

Transcriptional responses instigated by CCE, particularly *ilys-3* and *fmo-2*, reveal a distinct host response. However, we do not clearly indicate whether they

are part of a continued pattern-recognition response (PRR) to microbial-associated molecular patterns (MAMPs) or a general stress response perpetuated by damage-associated molecular patterns (DAMPs), or both. For instance, both MAMPs and DAMPs can promote initial immunity via autophagy but MAMP responses can also invoke general cellular stress that then propagate DAMP-mediated autophagy (Tang et al. 2012). In our case, it is likely that *E. faecalis* CCE initially promotes its colonization and protection from *S. aureus* with MAMPs but that superoxide-induced stress upon colonization promotes a DAMP response. Future work investigating the importance of *E. faecalis* CCE PRRs throughout the host response could more fully describe MAMP and DAMP importance and activity.

E. faecalis CCE effects on the C. elegans microbiome

My results indicate that exposure to *E. faecalis* CCE had no effect on subsequent microbiome assembly in terms of alpha diversity. In fact, out of all exposure treatments, only *P. mendocina* significantly affected alpha diversity, in which observed species diversity and the Chao 1 metric slightly decreased. In human systems (Chang et al. 2008), low Shannon diversity has been associated with adverse health outcomes, such as increased rates of necrotizing enterocolitis (McMurtry 2015). Changes in alpha diversity in nematodes have yet to be linked with health perturbations. Future work could address this and thus potentially address costs of defensive mutualists.

E. faecalis CCE also did not influence microbiome beta diversity different than other *E. faecalis* strains. However, exposure to the *E. faecalis* species in general, regardless of strain, had a slight impact on beta diversity compared to the *E. coli*

OP50 control and *P. mendocina*. This shows that pre-exposure to the *E. faecalis* species but not *E.* faecalis strains can minimally drive microbiome assembly. Convergence towards a "normal" microbiome regardless of early colonization is common in other hosts (Chu et al. 2017) (Nayfach et al. 2016).

E. faecalis CCE additionally had little effect on the assembly of other genera in the *C. elegans* microbiome. Importantly, *E. faecalis* CCE did not increase the abundance of any known *C. elegans* pathogens found in its surrounding soil. Additionally, it did not decrease the abundance of known *C. elegans* core microbiome members (Dirksen et al. 2016). Symbionts can have synergistic or antagonistic effects on other symbionts, effectively shifting symbiont services and costs (Doremus & Oliver 2017; Schwarz et al. 2016). For example, in honeybees, early exposure to a symbiont was linked with increased parasite colonization, a phenomenon that outweighed the symbionts benefits (Schwarz et al. 2016). Surprisingly, I observed that early exposure to *P. mendocina* and *E. faecalis* SE decreased the abundance of *Sphingomonas*, a known *C. elegans* core microbiome member. The degree to which this alters the overall benefit of these early exposures should be investigated.

I also observed that early exposure to all *E. faecalis* strains and *P. mendocina* significantly increased the abundance of a single RSV identified as *Enterococcus*. My microbiome analysis could not resolve species or strain level differences of this *Enterococcus*. This was limited by my sequencing approach (16S rRNA) and since the *E. faecalis* strains had no nucleotide differences in their 16S rRNA gene (King et al. 2016). Interestingly, *P. mendocina* does not limit infection by *E. faecalis* (Montalvo-Katz et al. 2013). It seems possible that this RSV was *E. faecalis*, but in

order to resolve that and its strain differences I would need to employ higher resolution sequencing (Kantor et al. 2017; Olm et al. 2017).

Even amongst a natural microbiome, *E. faecalis* CCE protected *C. elegans* better than other *E. faecalis* strains or the non-early exposure control. However, the level of protection by *E. faecalis* CCE amongst a microbiome was less than without a microbiome (King et al. 2016). Indeed, a dilution effect of *E. faecalis* amongst the microbiome is consistent with other systems showing that fitness constraints imparted by diverse interactions in polymicrobial communities can change (Gomez & Buckling 2011) and dilute (Sivan et al. 2015; Doremus & Oliver 2017) phenotypes normally observed in reduced systems. Nonetheless, *E. faecalis* CCE's sustained protection amongst a natural setting is promising since in some natural systems defensive microbes protective effects can be completely ameliorated (Lenhart & White 2017). This result suggests that *in vivo* experimentally evolved defensive microbes should be further explored for application in natural settings.

E. faecalis CCE's protective effect was directly correlated with early colonization abundance of *E. faecalis* strains. However, the degree to which the abundance of *E. faecalis* strains amongst a natural microbiomes relates to protection remains unresolved. Higher resolution sequencing technologies could be used to describe strain abundance as correlates with protection amongst a natural microbiome.

Future directions

The extent to which *E. faecalis* CCE evolved to modulate the *C. elegans* transcriptome is striking. Though extensive literature shows symbiont strains can specifically regulate host transcriptomes (Abt & Artis 2013; Cosseau et al. 2008; Park et al. 2016; Mandel et al. 2009), this work substantially adds that microbes can be experimentally evolved towards symbiosis to do so. Future research should explore other symbiont roles by evolving symbionts to influence diverse microbiome mediated-services such as nutrient acquisition (Rubino et al. 2017) and host development (Hosokawa 2016), and then similarly use RNASeq to describe regulated mechanisms. This work also shows that we can expand methods for yielding beneficial symbionts beyond isolation from existing microbiomes (Fujimura et al. 2014; Schwarzer et al. 2016) or genetic microbial engineering (Whitaker et al. 2017) to include experimental evolution. Such evolutionary engineering could have vast implications in applied fields. We showed these symbionts can evolve with minimal alterations on existing microbiota, at least at the broad community level. To ensure the lack of antagonistic effects on symbionts and their services, future research should integrate higher-resolution sequencing (Kantor et al. 2017; Olm et al. 2017). Evolved effects and diversification are often stronger and more rapid amongst increased selective pressure, multiple fitness peaks and increased genetic diversity (Martin & Wainwright 2013). Thus, future research could possibly increase protective effects by evolving protective microbes amongst higher phenotypic and genetic diversity, such as amongst a polymicrobial community. In all, this research shows some outcomes of experimentally evolved symbionts on

their host-microbiome system, and highlights experimentally evolved symbionts for potential utility for natural and applied contexts.

Methods

Strains

C. elegans used were Bristol N2, from *Caenorhabditis* Genetic Center. Bacterial *E. faecalis* strains were *E. faecalis* OG1RF (aka Anc) (Garsin et al. 2001), a isolate from the human gastrointestinal tract, and randomly selected *E. faecalis* SE and *E. faecalis* CCE from previously evolved lineages (King et al. 2016). *Pseudomonas mendocina* used was previously isolated from the natural *C. elegans* microbiome (Montalvo-Katz et al. 2013). I also used *S. aureus* strain MSSA476 (Holden et al. 2004), a disease-causing pathogen.

C. elegans exposures to E. coli OP50 and treatments

Culturing and *C. elegans* exposure of and to the *E. faecalis* Anc, *E. faecalis* SE, or *E. faecalis* CCE were the same as in King et al. (2016), with slight adjustments including a different washing procedure that was described by Ford et al. (2016). This procedure was confirmed to remove the majority of externally adhering bacteria by Berg et al. (2016). In short, this included removing cutaneous microbes by washing worms three times with M9 (Berg et al. 2016) over a filter tip and spinning at 800 g. In brief, for all experiments eggs were obtained from gravid worms by bleaching, approximately 1000 worms were exposed as L1s to *E. coli* OP50 at 20°C and allowed to develop for 24 h, then filter tip washed and transferred

to treatment exposures – the non-colonized exposure control *E. coli* OP50 (since *E. coli* OP50 are ground by the pharyngeal grinder and typically do not colonize *C. elegans* (Portal-Celhay & Blaser 2012)), *E. faecalis* strains (AE, SE OR CCE), or *P. mendocina* - at 25°C for 24 h. All bacteria were cultured overnight in lysogeny broth (LB) (*E. coli* OP50 and *P. mendocina*) or THB (*E. faecalis* strains and *S. aureus*), before being plated on NGM (*E. coli* OP50, 100ul) or TSA (*E. faecalis* strains, *P. mendocina*, *S. aureus*; all at 60ul) and cultured for 24 h at 30°C. Culture and exposure procedures were consistent in all assays (RNA extraction, soil exposure, gut accumulation, and protection persistence), with differences only in replicates, batch numbers and treatment exposures, and is now referred to as the standard experimental exposure. For the soil exposure experiment, worms were also early exposed to *P. mendocina*, which was cultured overnight in LB then, the same as *E. faecalis*, plated (60ul) on TSA and grown overnight at 30°C. For my *E. faecalis* SE and *E. faecalis* CCE strains, I randomly selected lineages from the previous evolution experiment (King et al. 2016). The same evolved lineages were used for all batches. For *E. faecalis* CCE this was lineage CCE-A and for *E. faecais* SE it was SE-A. Throughout the experiments, for cultures and plating of all treatments, each colony was twice streaked to ensure that they were isogenic.

RNA extraction and library preparation

Four replicates of each treatment were prepared for RNA extractions using the standard experimental exposure, where treatments were *E. coli* OP50 (control) and *E. faecalis* strains (AE, SE or CCE). Approximately 500 worms were used for each RNA extraction. To clean the outside of *C. elegans* after treatment exposures, I

used a gravity washing procedure. This is different than the other experiments, in which filter tip washing was continuously used through the experiment. In brief, worms were suspended in M9 buffer and allowed to gravity pellet, then removed and transferred to clean M9 for a total of three times. I extracted RNA by adding 1ml TRIzol (Invitrogen) followed by three iterations of freeze-thawing with liquid nitrogen and added 200ul chloroform/mL TRIzol. The mixture was then centrifuged at 4°C for 15min at 12,000 g. The upper aqueous phase was then supplemented with 1 volume ethanol (100%). The mixture was then transferred to a Zymo-Spin IC Column (Zymo) and centrifuged for 30 seconds at 15,600 g. I added 400ul RNA wash buffer (Zymo) to the column, centrifuged the samples at 15,600 g and treated them with DNase digestion mix (1:7 DNase I: DNA Digestion Buffer) (Zymo). Following this, I added 400ul RNA Prep Buffer (Zymo) and centrifuged for 30 seconds at 15,600 g. I then washed the RNA twice with RNA Wash Buffer (Zymo) and eluted the RNA in 30ul DNase-free water. Library preparations and 75bp paired-end sequencing on the HiSeq4000 were conducted by The High-Throughput Genomics Group at the Wellcome Trust Center for Human Genetics.

Compost preparation

Overripe bananas were supplemented to Westland Multi-Purpose Compost with added John Innes (Westland Horticulture; Dungannon, UK) to enrich microbiota via carbohydrates and left to compost at 20°C for 5 days before disrupted and washed to create a microbial extract. To create the microbial wash, I added 2ml M9 to 5 g compost in a 50ml conical tube, vortexed vigorously for 60 seconds, transferred a 10ml aliquot to a 15ml conical tube and centrifuged the

mixture for one minute at 300 g, and created a glycerol stock (25%) of the wash that was immediately stored at -80°C. To reconstitute compost with microbes prior to worm addition, 5g of autoclaved compost was supplemented with 1ml microbial wash and incubated for 48h at 25°C prior to addition of worms (Berg et al. 2016).

Worm compost exposure and harvesting

Five replicates of each treatment repeated over three replicate batches were used for compost exposures. Following the standard treatment exposure - where treatments were *E. coli* OP50 (control), *E. faecalis* strains (AE, SE or CCE), or *P. mendocina* - worms were extensively filter tip washed and transferred to microbial enriched soil for 24h, after which ~700 worms were harvested over 2h using a Baermann funnel lined with tissue paper (Barriere 2006), then filter tip washed again and immediately stored at -80°C until DNA extractions.

DNA extractions

gDNA was isolated from compost exposed worms (~700) or soil (0.25g) using the MO BIO PowerSoil DNA Isolation Kit (12888; MO BIO Laboratories; Carlsbad, CA, USA), with slight adjustments. For homogenization and cell lysis, I attached the MO BIO kit's PowerBead Tubes to the Benchmark Scientific BeadBlaster Homogenizer (D1030-E; Benchmark Scientific; South Plainfield, NJ, USA) and homogenized and lysed cells for 60 seconds at 2800 rpm. Final gDNA was released from the silica membrane using 40ul sterile, nuclease-free water (Promega; Madison, WI, USA).

16S rRNA library preparation

The 16S rRNA V4 region was amplified from the worm microbiome gDNA using the 515F Golay-barcoded primers and 806R, primers revised by by Apprill et al. and developed by Caporaso et al . (Caporaso et al. 2012; Apprill et al. 2015) and listed on the Earth Microbiome Project (EMP) 16S protocol site (http://www.earthmicrobiome.org/emp-standard-protocols/16s/). Samples were prepared in accordance with the standard EMP 16S rRNA protocol. 25ul polymerase-chain reactions (PCR) contained 10ul Platinum Hot Start MM (2X) (company), 11ul nuclease-free water, 1 ul of each forward and reverse primer (0.20 uM final concentrations), and 2ul gDNA template. No-template controls (NTCs) contained nuclease free water in lieu of gDNA. Reactions were held at 94°C for 3min to denature the DNA, and amplification took place for 35 cycles at 94°C for 45 sec, 50°C for 60 sec and, 72°C for 90 sec. The cycles were followed by a hold at 72°C for 10 min. Amplicons were visualized on a 1.5% agarose gel. gDNA was quantified using the Qubit 2.0 (Thermofisher, Bartlesville, OK) and amplicons were pooled at equimolar ratios (~ 240ng per sample). The combined amplicon pool was then cleaned using the Qiagen PCR Purification Kit (Qiagen, Germantown, MD). The multiplexed library was quality checked and sequenced with the MiSeq 2x250nt PE v2 protocol at the W.M. Keck Center for Comparative and Functional Genomics (University of Illinois at Urbana-Champaign; Urbana, IL, USA).

Gut accumulation enumeration and protection persistence

Five replicates of each treatment from the same batch were used for gut accumulation enumeration and protection persistence assays. Following the standard treatment exposure - where treatments were *E. coli* OP50 (control), *E. faecalis* strains (AE, SE or CCE), or *P. mendocina* - worms were extensively filter tip washed and then either transferred to microcentrifuge tubes containing ten 1 mm

zirconia/silica beads in 50ul M9, for the gut accumulation enumeration, or advanced to soil exposures for the protection persistence assay. For gut accumulation enumerations, the worms were homogenized and gut bacteria released using the Benchmark Scientific BeadBlaster Homogenizer (D1030-E; Benchmark Scientific; South Plainfield, NJ, USA) for 45 seconds at 2800 rpm. Dilution series of the mixture were plated on TSA and CFUs were enumerated after incubating at 30°C for 24 h. For the protection persistence assay worms were transferred to plates with *S. aureus* and exposed for 24 h at 25°C. After exposure, I calculated mortality by counting alive and dead worms. For plotting and statistical analyses, I have provided an R markdown file outlining my analyses of gut CFU and protection data (Supplementary file 1).

RNASeq bioinformatic processing and analyses

To summarize my RNASeq bioinformatic workflow, I provide a flow chart outlining methods (Supplementary Figure 8). In short, I trimmed and filtered reads using Trimmomatic (Bolger et al. 2014), pseudoaligned reads and quantified abundances of transcripts using kallisto (Bray et al. 2016), and conducted differential expression analyses using sleuth (Pimentel et al. 2016). For the Trimmomatic and kallisto steps conducted in Linux, I provide a supplementary workflow file using the MIT-licensed workflow manager Snakemake (snakemake.readthedocs.io) (Supplementary File 2), and for my sleuth analysis conducted using R (3.4.0) I provide a fully reproducible workflow in an R markdown file (Supplementary file 3). Other R libraries used include ggplot2 (Wickham 2009),

devtools, biomaRt (Durinck et al. 2005), VennDiagram (Chen & Boutros 2011) and GOplot (Walter et al. 2015) along with their dependencies.

16S rRNA bioinformatic processing and analyses

PhiX sequences were first removed from my library using Bowtie2 by mapping my reads against an index built from a phiX genome (found at support.illumina.com/sequencing/sequencing_software/igenome.html). Demultiplexed, paired-end fastq files were then processed in R (3.4.0) using DADA2 (Callahan et al. 2016) as previously described (Callahan et al. 2016). In short, this included filtering and trimming, error rate estimation, dereplication of reads into unique sequences, and ribosomal variant inference. I then merged paired-end reads, constructed a ribosomal sequence variant (RSV) table (sample x sequence abundance matrix), and removed chimeras. I also used DADA2's native implementation of the Ribosomal Database Project (RDP) naïve Bayesian classifier (Cole et al. 2013) trained against the GreenGenes 13.8 release reference fasta (https://zenodo.org/record/158955#.WQsM81Pyu2w) to classify RSVs taxonomically. For DADA2 processing I provide a reproducible R Markdown file (Supplementary file 4).

For differential abundance of taxa analyses I corrected for batch effects by incorporating batch as a term in the design formula of my DESeq2 analysis (Supplementary file 4). For alpha diversity analyses I rarefied to an even sampling depth of 22,873 reads per sample. To calculate beta diversity I built a distance matrix based on samples' weighted UniFrac scores (Lozupone & Knight 2005), and performed PCoA on the distance matrix. And, to represent high-level beta diversity

between microbial communities influenced by treatment, I filtered out lowly observed and lowly abundant RSVs and removed a batch effect after stabilizing for variance.

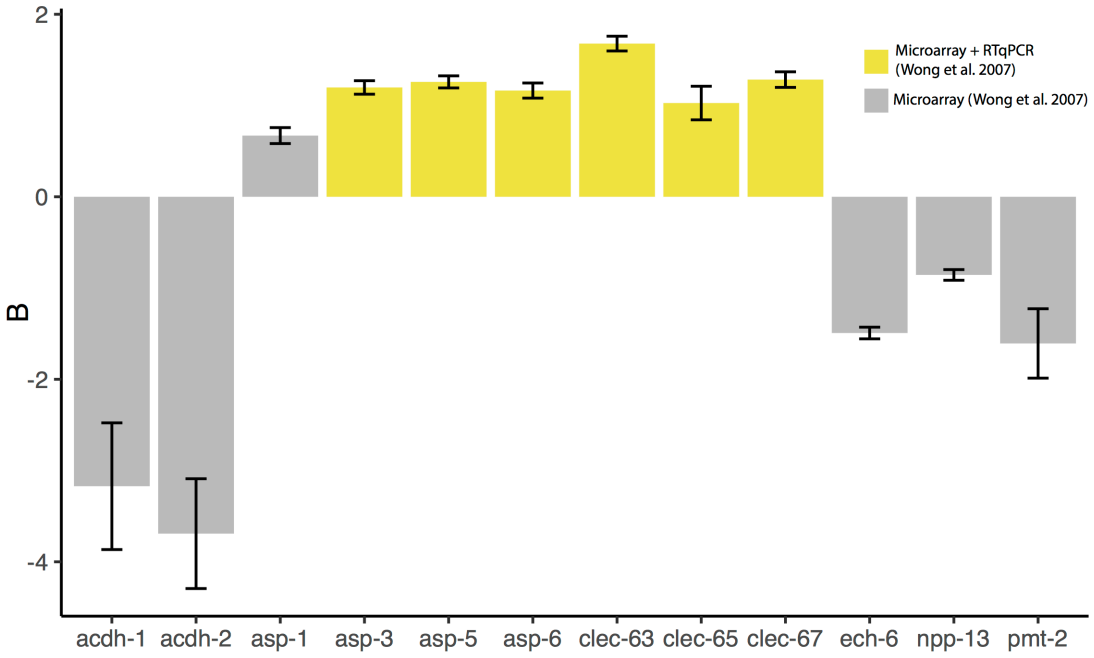
I created visualizations and conducted statistical analyses on the RSV table in R (3.4.0). To calculate alpha diversity measurements of observed RSVs, Shannon's index and Chao 1, I used phyloseq's (1.16.2) (McMurdie & Holmes 2013) estimate_richness function. Phyloseq was also used to perform ordinations, using Principle Coordinate Analysis (PCoA) on UniFrac distance scores (Lozupone & Knight 2005). To perform differential abundances analyses I used the DESeq2 package to estimate differential abundance based on a negative binomial distribution (Love et al. 2014). Other R packages used include: ggplot2, for visualizing data and making figures (2.0.0) (Wickham 2009); Rcpp for C++ parallelization in R (Eddelbuettel & François 2011); optparse (1.3.2.) to parse command line options; stats (3.2.3) to conduct statistics; and data.table (1.9.6) to handle data frames. For my 16S rRNA analyses I have also provided an R markdown file outlining a fully reproducible workflow (Supplementary file 4).

Code availability

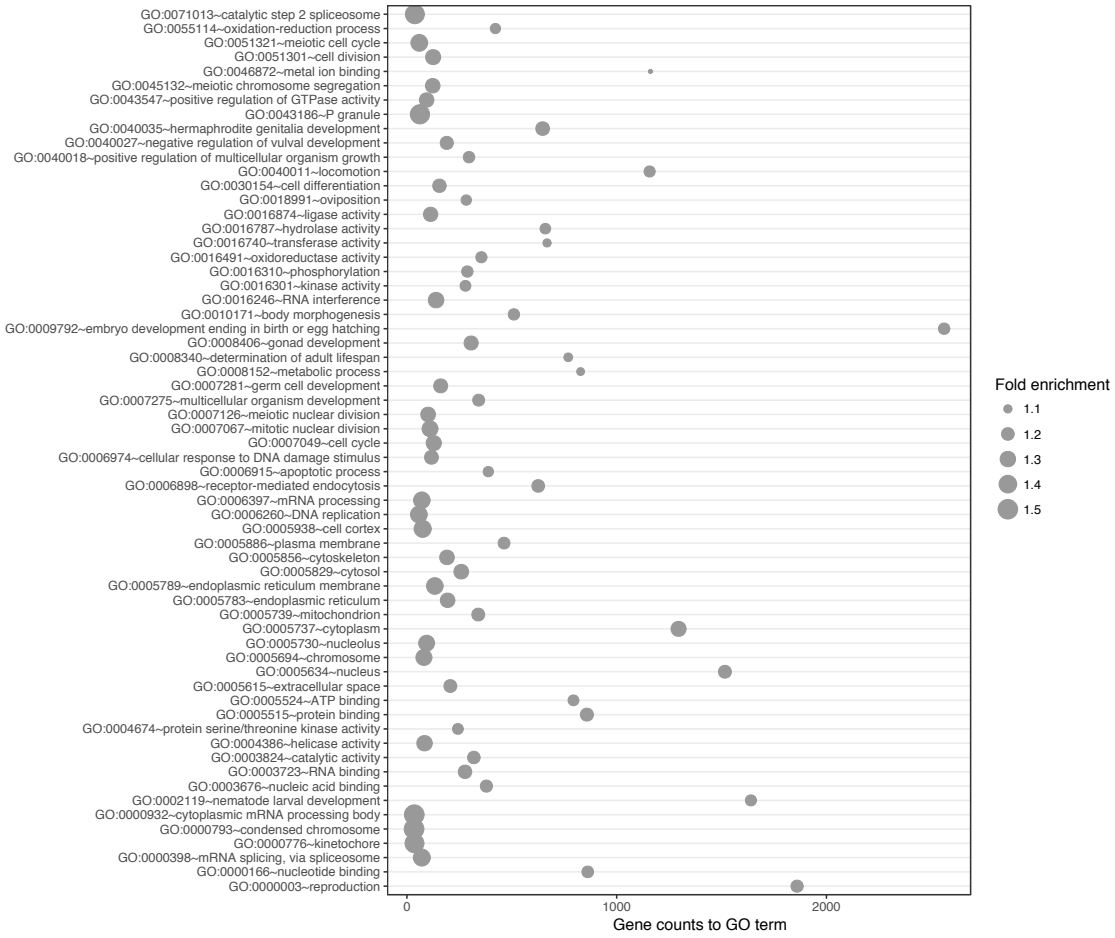
The packages and pipelines used are available, with documentation, on their respective sites and repositories. Concerning the main pipelines used, kallisto (https://pachterlab.github.io/kallisto/), sleuth (https://pachterlab.github.io/sleuth/), DADA2 (https://github.com/benjjneb/dada2), and phyloseq (https://joey711.github.io/phyloseq/) are all open-source and publicly available. R

markdown files for implementing these packages on my data are available in supplementary files (Supplementary files 2-4).

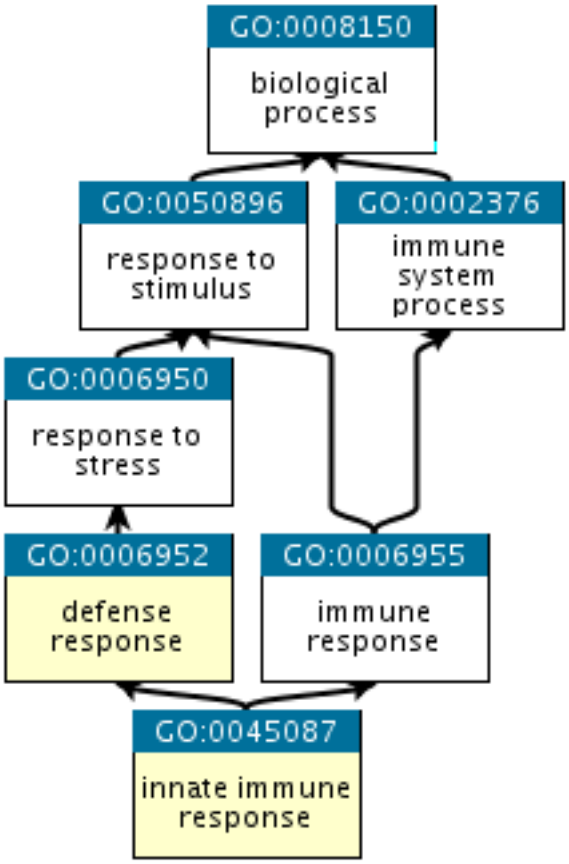
Supplementary Figures



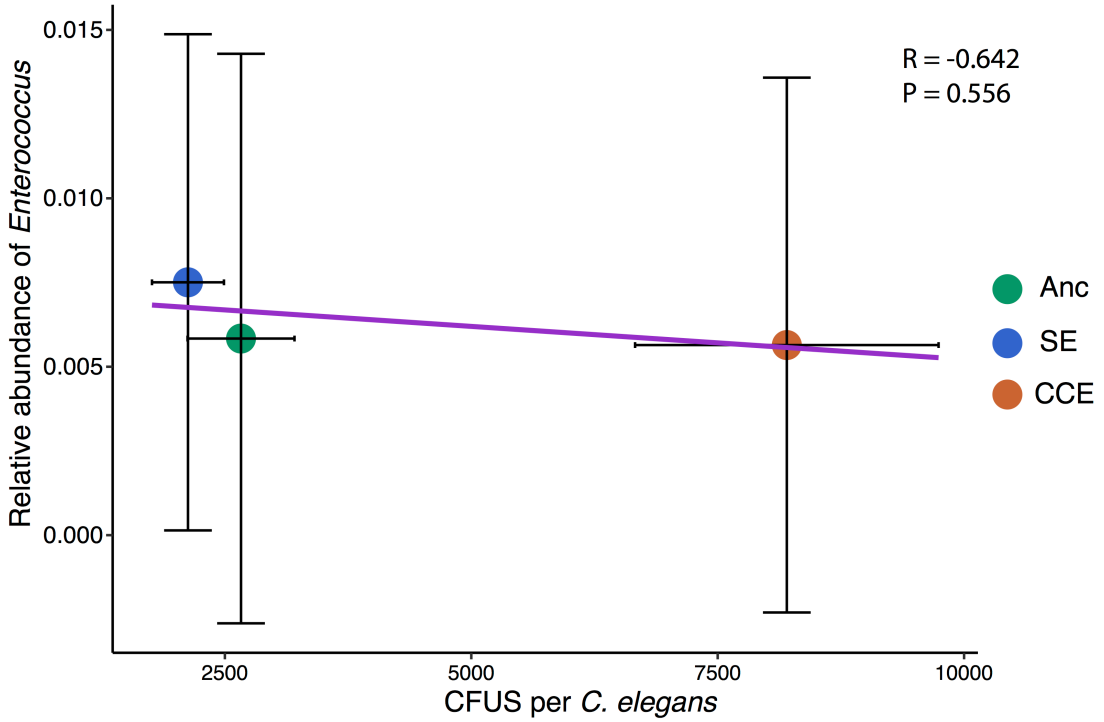
Supplementary Figure 1. Differential gene expression of previously investigated genes. *C. elegans* DEGs related to pathogenesis and *E. faecalis* colonization. Six genes (yellow) previously assayed with microarray and confirmed using RTqPCR, and others (grey) were only assayed with microarray (Wong et al. 2007). Showing β-values from using a Wald-test (adj-P < 0.05). With the exception of npp-13, all genes agreed in direction (up or down) of differential expression.



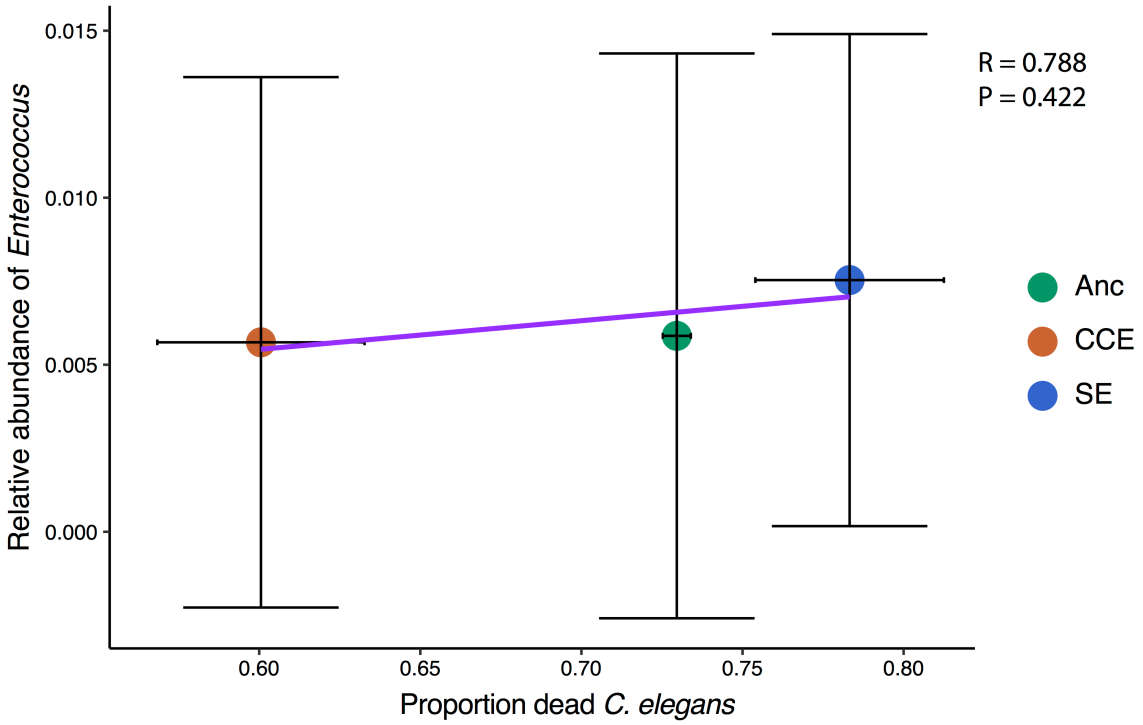
Supplementary Figure 2. *E. faecalis* OG1RF gene counts to GO term enrichment. Showing GO terms significantly enriched in *C. elegans* exposed to *E. faecalis* OG1RF compared to *C. elegans* exposed to *E. coli* OP50 (DAVID 6.8; 2016 build; adj-P < 0.05). DEGs from sleuth Wald-Test (adj-P; < 0.05).



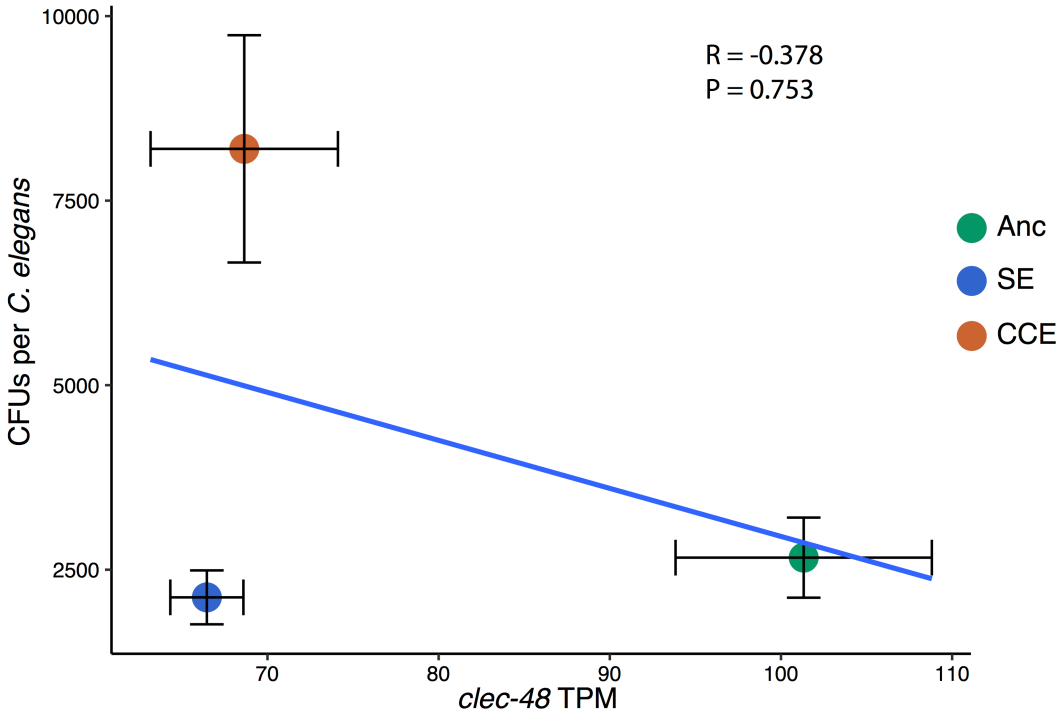
Supplementary Figure 3. GO term ancestor chart from *E. faecalis* CCE to *E. faecalis* SE comparison. Highlight depicts enriched GO terms. Innate immune response is a part of defense response. Generated using EMBL-EBI QuickGO beta.



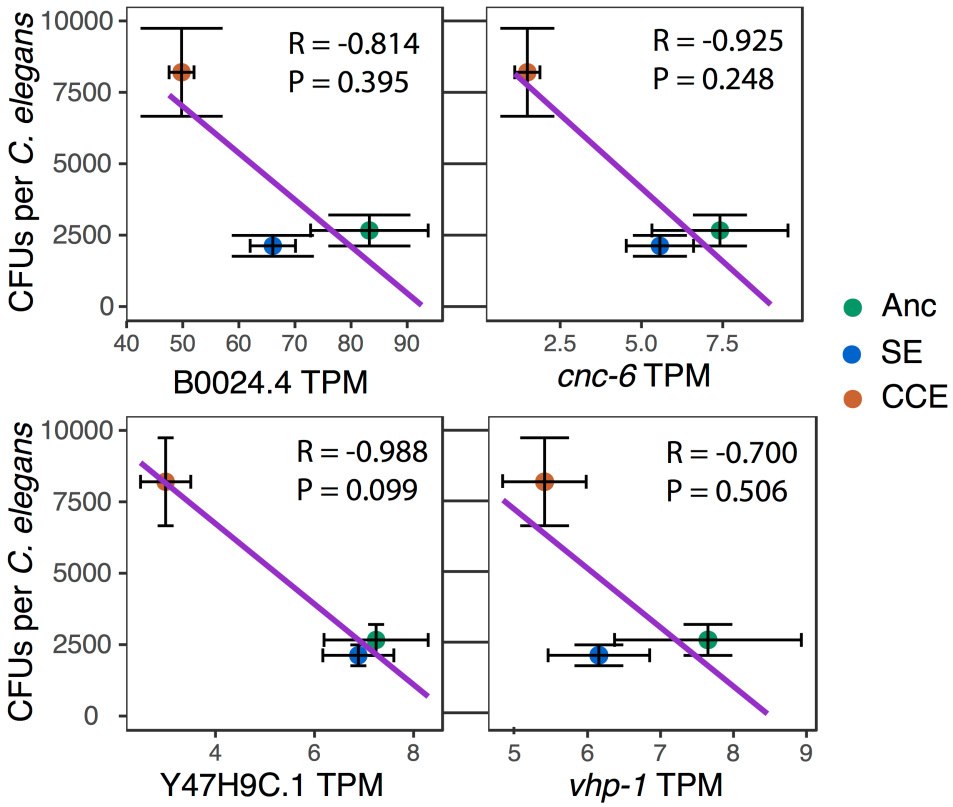
Supplementary figure 4. *E. faecalis* CFUs in *C. elegans* and relative abundance of *Enterococcus* amongst microbiome. X-axis is *C. elegans* gut bacterial CFUs after exposure to *E. faecalis* Anc, *E. faecalis* SE, or *E. faecalis* CCE. Y-axis is relative abundance of *Enterococcus* in *C. elegans* amongst microbiome. There was no significant correlation between *E. faecalis* CFUs and *Enterococcus* relative abundance (Pearson's; R = -0.642; P = 0.556). Error bars = ± s.e. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved.



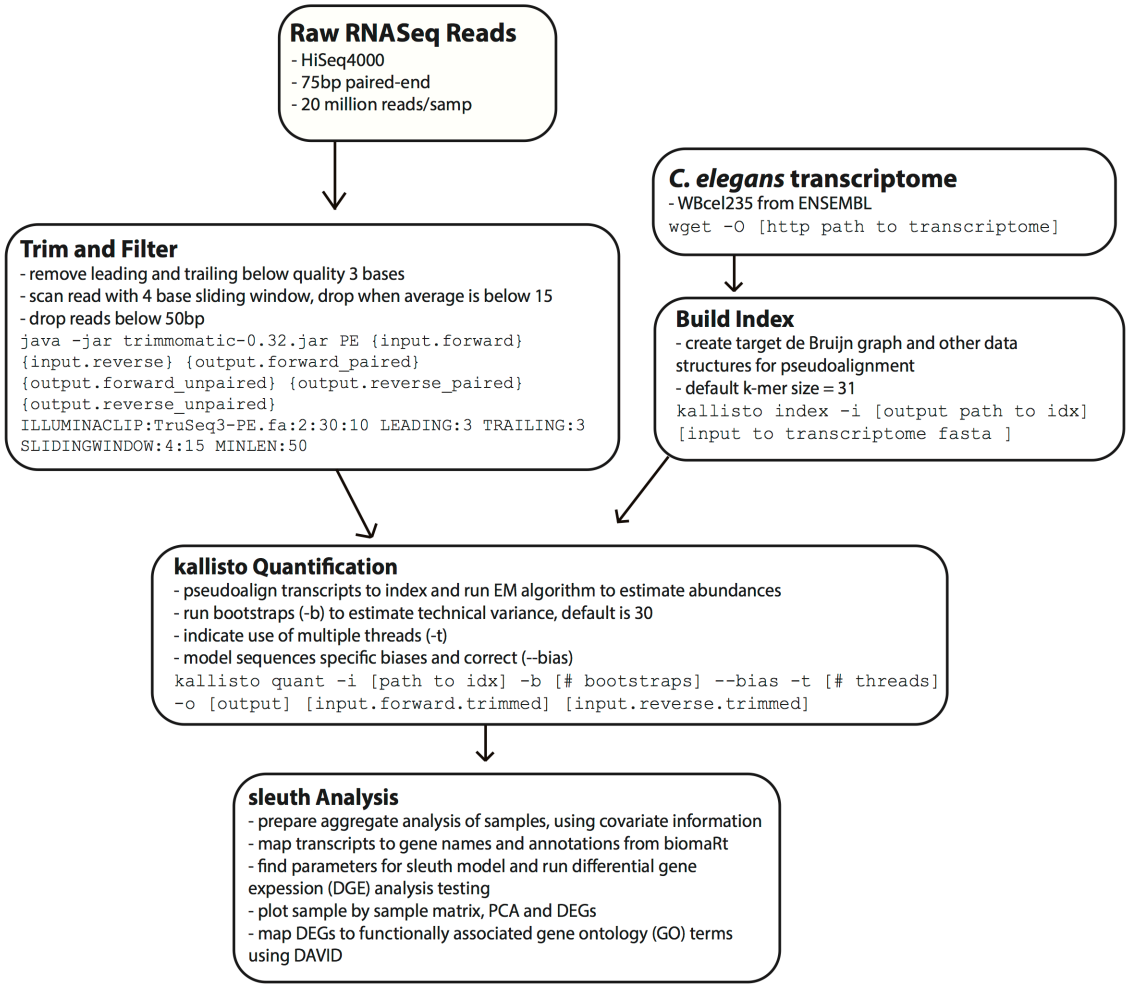
Supplementary figure 5. Relative abundance of *Enterococcus* in microbiome and proportion dead *C. elegans*. X-axis is proportion dead *C. elegans* after *S. aureus* exposure. Y-axis is relative abundance of *Enterococcus* in *C. elegans* amongst microbiome. There was no significant correlation between *E. faecalis* CFUs and *Enterococcus* relative abundance (Pearson's; R = 0.788; P = 0.422). Error bars = ± s.e. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved.



Supplementary figure 6. Correlation between *E. faecalis* CFUs in *C. elegans* guts and *clec-48* TPM values. *C. elegans* gut bacterial CFUs after exposure to *E. faecalis* Anc, *E. faecalis* SE, or *E. faecalis* CCE correlated with transcript per million (TPM) values for transcripts identified as *ilys-3* from RNASeq experiments. *clec-48* abundance is not a predictor of *E. faecalis* CFUs (Pearson's; R = -0.378; P = 0.753). Data for CFUs and transcript levels collected at the same time points but from different batches and are means. CFUs collected from n = 5 replicate populations per treatment. RNASeq collected from n = 4 replicate populations per treatment. Error bars = ± s.e.. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved.



Supplementary figure 7. Correlation between *E. faecalis* CFUs in *C. elegans* guts and downregulated immune-related transcripts. *C. elegans* gut bacterial colony forming units (CFUs) after exposure to *E. faecalis* Anc, *E. faecalis* SE, or *E. faecalis* CCE correlated with TPM values for transcripts identified as downregulated in *E. faecalis* CCE treatment and associated to innate immune response or defense response GO terms. None were significant predictors of CFUs in *C. elegans*. Data for CFUs and transcript levels collected at the same time points but from different batches and are means. CFUs collected from n = 5 replicate populations per treatment. RNASeq collected from n = 4 replicate populations per treatment. Error bars = ± s.e.. Anc = *E. faecalis* ancestor. SE = *E. faecalis* single-evolved. *E. faecalis* CCE = *E. faecalis* co-colonized evolved.



Supplementary figure 8. RNASeq bioinformatic workflow

Supplementary Tables

Supplementary table 1. 16653 DEGs from *E. faecalis* Anc exposure.
https://www.dropbox.com/s/icng7v4o8tew81g/supp_table1_ef_op50_degs.csv?dl=0

Supplementary table 2. List of DEGs from *C. elegans* exposures to *E. faecalis* CCE and *E. faecalis* SE compared to *E. faecalis* Anc exposure.
https://www.dropbox.com/s/lf9ka8lgr5j68x1/supp_table2_se_cce_ac_degs.csv?dl=0

Supplementary table 3. List of GO terms and associated DEGs comparing *C. elegans* exposures to *E. faecalis* SE and *E. faecalis* CCE compared to *E. faecalis* Anc.
https://www.dropbox.com/s/x0k6y0072jr96j4/supp_table3_cce_se_go.xlsx?dl=0

Supplementary table 4. List of DEGs from *C. elegans* exposure to *E. faecalis* CCE compared to *E. faecalis* Anc.
https://www.dropbox.com/s/r4lcvuy8k3401c5/supp_table4_cce_se_degs.csv?dl=0

Supplementary table 5. ANOVA and Tukey-HSD tables for model for the affect of batch and treatment on observed RSVs.

	DF	SS	MS	F	P
Batch	1	283	283	1.37	0.249
Treatment	4	3182	795	3.84	0.010
Residuals	39	8077	207		

		Tukey HSD		
	diff	**lwr**	**upr**	**adj-P**
CCE-Anc	11.5	-6.90	29.9	0.396
OP50-Anc	12.7	-9.81	35.3	0.498
Pm-Anc	-10.3	-28.7	8.10	0.506
SE-Anc	5.6	-12.8	24.0	0.906
OP50-CCE	1.23	-21.3	23.8	1.00
Pm-CCE	-21.8	-40.2	-3.40	0.013
SE-CCE	-5.9	-24.3	12.5	0.889
Pm-OP50	-23.0	-45.6	-0.485	0.043
SE-OP50	-7.13	-29.7	15.4	0.894
SE-Pm	15.9	-2.50	34.3	0.118

Supplementary table 6. ANOVA and Tukey-HSD tables for model for the affect of batch and treatment on Chao 1 diversity.

	DF	SS	MS	F	P
Batch	1	214	214	0.938	0.339
Treatment	4	3356	839	3.67	0.012
Residuals	39	8907	228		

	Tukey HSD			
	diff	**lwr**	**upr**	**adj-P**
CCE-Anc	11.9	-7.40	31.3	0.408
OP50-Anc	13.5	-10.2	37.2	0.487
Pm-Anc	-10.3	-29.6	9.02	0.553
SE-Anc	5.84	-13.5	25.2	0.908
OP50-CCE	1.57	-22.1	25.2	1.00
Pm-CCE	-22.2	-41.6	-2.91	0.017
SE-CCE	-6.09	-25.4	13.2	0.895
Pm-OP50	-23.8	-47.5	-0.138	0.048
SE-OP50	-7.66	-31.3	16.0	0.885
SE-Pm	16.1	-3.18	35.5	0.140

Supplementary table 7.
https://www.dropbox.com/s/xwpv71z1obt9bpa/supp_table7_all_deb.csv?dl=0

Bibliography

Abt, M.C. & Artis, D., 2013. The dynamic influence of commensal bacteria on the immune response to pathogens. *Current Opinion in Microbiology*, 16(1), pp.4–9.

Alper, S. et al., 2007. Specificity and Complexity of the Caenorhabditis elegans Innate Immune Response. *Molecular and cellular biology*, 27(15), pp.5544–5553.

Apprill, A. et al., 2015. Minor revision to V4 region SSU rRNA 806R gene primer greatly increases detection of SAR11 bacterioplankton. *Aquatic Microbial Ecology*, 75(2), pp.129–137.

Barriere, A., 2006. Isolation of C. elegans and related nematodes. *WormBook*, pp.1–9.

Baumann, P. et al., 1995. Genetics, Physiology, and Evolutionary Relationships of the Genus Buchnera: Intracellular Symbionts of Aphids. *Annual Review of Microbiology*, 49(1), pp.55–94.

Bäumler, A.J. & Sperandio, V., 2016. Interactions between the microbiota and pathogenic bacteria in the gut. *Nature*, 535(7610), pp.85–93.

Becker, M.H. et al., 2009. The Bacterially Produced Metabolite Violacein Is Associated with Survival of Amphibians Infected with a Lethal Fungus. *Applied and environmental microbiology*, 75(21), pp.6635–6638.

Berg, M. et al., 2016. Assembly of the Caenorhabditis elegans gut microbiota from diverse soil microbial environments. pp.1–12.

Blaxter, M.L. et al., 1992. Nematode surface coats: Actively evading immunity. *Parasitology Today*, 8(7), pp.243–247.

Boeck, M.E. et al., 2016. The time-resolved transcriptome of C. elegans. *Genome Research*, 26(10), pp.1441–1450.

Bolger, A.M., Lohse, M. & Usadel, B., 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15), pp.2114–2120.

Bray, N.L. et al., 2016. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5), pp.525–527.

Broxton, C.N. & Culotta, V.C., 2016. SOD Enzymes and Microbial Pathogens: Surviving the Oxidative Storm of Infection D. C. Sheppard, ed. *PLoS Pathogens*, 12(1), pp.e1005295–6.

Brucker, R.M. et al., 2008. Amphibian Chemical Defense: Antifungal Metabolites of the Microsymbiont Janthinobacterium lividum on the Salamander Plethodon cinereus. *Journal of Chemical Ecology*, 34(11), pp.1422–1429.

Buffie, C.G. & Pamer, E.G., 2013. Microbiota-mediated colonization resistance against intestinal pathogens. pp.1–12.

C. elegans Sequencing Consortium, 1998. Genome sequence of the nematode C. elegans: a platform for investigating biology. *Science*, 282(5396), pp.2012–2018.

Cabreiro, F. & Gems, D., 2013. Worms need microbes too: microbiota, health and aging in Caenorhabditis elegans. *EMBO Molecular Medicine*, 5(9), pp.1300–1310.

Callahan, B.J. et al., 2016. DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods*, pp.1–7.

Caporaso, J.G. et al., 2012. Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *The ISME journal*, 6(8), pp.1621–1624.

Chang, J.Y. et al., 2008. Decreased diversity of the fecal Microbiome in recurrent Clostridium difficile-associated diarrhea. *The Journal of Infectious Diseases*, 197(3), pp.435–438.

Chen, H. & Boutros, P.C., 2011. VennDiagram: a package for the generation of highly-customizable Venn and Euler diagrams in R. *BMC Bioinformatics*, 12(1), p.35.

Chu, D.M. et al., 2017. Maturation of the infant microbiome community structure and function across multiple body sites and in relation to mode of delivery. *Nature Medicine*, 23(3), pp.314–326 13.

Chung, H. et al., 2012. Gut Immune Maturation Depends on Colonization with a Host-Specific Microbiota. *Cell*, 149(7), pp.1578–1593.

Clark, L.C. & Hodgkin, J., 2013. Commensals, probiotics and pathogens in the Caenorhabditis elegansmodel. *Cellular Microbiology*, 16(1), pp.27–38.

Cole, J.R. et al., 2013. Ribosomal Database Project: data and tools for high throughput rRNA analysis. *Nucleic Acids Research*, 42(D1), pp.D633–D642.

Cosseau, C. et al., 2008. The Commensal Streptococcus salivarius K12 Downregulates the Innate Immune Responses of Human Epithelial Cells and Promotes Host-Microbe Homeostasis. *Infection and Immunity*, 76(9), pp.4163–4175.

Dirksen, P. et al., 2016. The native microbiome of the nematode Caenorhabditis elegans: gateway to a new host-microbiome model. *BMC Biology*, pp.1–16.

Doremus, M.R. & Oliver, K.M., 2017. Aphid Heritable Symbiont Exploits Defensive Mutualism H. L. Drake, ed. *Applied and Environmental Microbiology*, 83(8), pp.e03276–16–45.

Doublet, V. et al., 2017. Unity in defence: honeybee workers exhibit conserved molecular responses to diverse pathogens. pp.1–17.

Durinck, S. et al., 2005. BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16), pp.3439–3440.

Eddelbuettel, D. & François, R., 2011. Rcpp: Seamless Rand C++Integration. *Journal of Statistical Software*, 40(8).

Félix, M.-A. & Braendle, C., 2010. The natural history of Caenorhabditis elegans. *Current Biology*, 20(22), pp.R965–R969.

Ford, S.A. et al., 2016. Microbe-mediated host defence drives the evolution of reduced pathogen virulence. *Nature Communications*, 7, pp.1–9.

Fritz, J.V. et al., 2013. From meta-omics to causality: experimental models for human microbiome research. *Microbiome*, 1(1), p.14.

Fuentes, S. et al., 2014. Reset of a critically disturbed microbial ecosystem: faecal transplant in recurrent Clostridium difficile infection. 8(8), pp.1621–1633.

Fujimura, K.E. et al., 2014. House dust exposure mediates gut microbiome Lactobacillus enrichment and airway immune defense against allergens and virus infection. *Proceedings of the National Academy of Sciences*, 111(2), pp.805–810.

Garsin, D.A. et al., 2001. A simple model host for identifying Gram-positive virulence factors. *Proceedings of the National Academy of Sciences*, 98(19), pp.10892–10897.

Gerardo, N.M. et al., 2010. Immunity and other defenses in pea aphids, Acyrthosiphon pisum. *Genome Biology*, 11(2), pp.R21–17.

Gomez, P. & Buckling, A., 2011. Bacteria-Phage Antagonistic Coevolution in Soil. *Science*, 332(6025), pp.106–109.

Gravato-Nobre, M.J. et al., 2016. The Invertebrate Lysozyme Effector ILYS-3 Is Systemically Activated in Response to Danger Signals and Confers Antimicrobial Protection in C. elegans D. S. Schneider, ed. *PLoS Pathogens*, 12(8), pp.e1005826–42.

Gray, J.C. & Cutter, A.D., 2014. Mainstreaming Caenorhabditis elegans in experimental evolution. *Proceedings. Biological sciences / The Royal Society*, 281(1778), pp.20133055–20133055.

Hall, J.P.J. et al., 2016. Source-sink plasmid transfer dynamics maintain gene mobility in soil bacterial communities. *Proceedings of the National Academy of Sciences of the United States of America*, 113(29), pp.8260–8265.

Han, L. et al., 2016. The relationships among host transcriptional responses reveal distinct signatures underlying viral infection-disease associations. *Molecular*

BioSystems, 12, pp.653–665.

Hoang, K.L., Morran, L.T. & Gerardo, N.M., 2016. Experimental Evolution as an Underutilized Tool for Studying Beneficial Animal–Microbe Interactions. *Frontiers in Microbiology*, 07(e1004182), pp.109–16.

Holden, M.T.G. et al., 2004. Complete genomes of two clinical Staphylococcus aureus strains: Evidence for the rapid evolution of virulence and drug resistance. *Proceedings of the National Academy of Sciences*, 101(26), pp.9786–9791.

Hosokawa, T., 2016. Obligate bacterial mutualists evolving from environmental bacteria in natural insect populations. *Nature Microbiology*, 1(1), pp.1

–7

.

Howe, K.L. et al., 2016. WormBase 2016: expanding to enable helminth genomic research. *Nucleic Acids Research*, 44(D1), pp.D774–D780.

Hrček, J., McLean, A.H.C. & Godfray, H.C.J., 2016. Symbionts modify interactions between insects and natural enemies in the field P. Amarasekare, ed. *Journal of Animal Ecology*, 85(6), pp.1605–1612.

Irazoqui, J.E. et al., 2010. Distinct Pathogenesis and Host Responses during Infection of C. elegans by P. aeruginosa and S. aureus D. S. Guttman, ed. *PLoS Pathogens*, 6(7), pp.e1000982–24.

Kantor, R.S. et al., 2017. Genome-Resolved Meta-Omics Ties Microbial Dynamics to Process Performance in Biotechnology for Thiocyanate Degradation. *Environmental Science & Technology*, 51(5), pp.2944–2953.

Kim, S.-H. & Lee, W.-J., 2014. Role of DUOX in gut inflammation: lessons from Drosophila model of gut-microbiota interactions. *Frontiers in cellular and infection microbiology*, 3, pp.1–12.

King, K.C. et al., 2016. Rapid evolution of microbe-mediated protection against pathogens in a worm host. *The ISME Journal*, pp.1–10.

Kremer, N. et al., 2013. Initial Symbiont Contact Orchestrates Host-Organ-wide Transcriptional Changes that Prime Tissue Colonization. *Cell host & microbe*, 14(2), pp.183–194.

LaMunyon, C.W., Bouban, O. & Cutter, A.D., 2006. Postcopulatory Sexual Selection Reduces Genetic Diversity in Experimental Populations of Caenorhabditis elegans. *Journal of Heredity*, 98(1), pp.67–72.

Lee, K.-A. et al., 2015. Bacterial Uracil Modulates Drosophila DUOX- Dependent Gut Immunity via Hedgehog-Induced Signaling Endosomes. *Cell host & microbe*,

17(2), pp.191–204.

Lee, K.-H. & Ruby, E.G., 2004. Competition between Vibrio fischeri Strains during Initiation and Maintenance of a Light Organ Symbiosis. *Journal of bacteriology*, 179, pp.1985–1992.

Lenhart, P.A. & White, J.A., 2017. A defensive endosymbiont fails to protect aphids against the parasitoid community present in the field. *Ecological Entomology*, 39, p.736.

Love, M.I., Huber, W. & Anders, S., 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12), pp.31–21.

Lozupone, C. & Knight, R., 2005. UniFrac: a New Phylogenetic Method for Comparing Microbial Communities. *Applied and Environmental Microbiology*, 71(12), pp.8228–8235.

Mallo, G.V. et al., 2002. Inducible antibacterial defense system in C. elegans. *Current Biology*, 12(14), pp.1209–1214.

Mandel, M.J. et al., 2009. A single regulatory gene is sufficient to alter bacterial host range. *Nature (News Feature)*, 457(7235), pp.215–218.

Marcobal, A. et al., 2013. A metabolomic view of how the human gut microbiota impacts the host metabolome using humanized and gnotobiotic mice. 7(10), pp.1933–1943.

Martin, C.H. & Wainwright, P.C., 2013. Multiple Fitness Peaks on the Adaptive Landscape Drive Adaptive Radiation in the Wild. *Science*, 339(6116), pp.208–211.

McCallum, K.C. & Garsin, D.A., 2016. The Role of Reactive Oxygen Species in Modulating the Caenorhabditis elegans Immune Response J. M. Leong, ed. *PLoS Pathogens*, 12(11), pp.e1005923–6.

McMurdie, P.J. & Holmes, S., 2013. phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data M. Watson, ed. *PLoS ONE*, 8(4), pp.e61217–11.

McMurtry, V.E., 2015. Bacterial diversity and Clostridia abundance decrease with increasing severity of necrotizing enterocolitis. pp.1–8.

Moeller, A.H. et al., 2016. Cospeciation of gut microbiota with hominids. *Science*, 353(6297), pp.380–382.

Montalvo-Katz, S. et al., 2013. Association with soil bacteria enhances p38-dependent infection resistance in Caenorhabditis elegans. *Infection and Immunity*, 81(2), pp.514–520.

Morran, L.T. et al., 2016. Nematode-bacteria mutualism: Selection within the mutualism supersedes selection outside of the mutualism. *Evolution*, 70(3), pp.687–695.

Morran, L.T. et al., 2011. Running with the Red Queen: Host-Parasite Coevolution Selects for Biparental Sex. *Science*, 333(6039), pp.216–218.

Nakatsuji, T. et al., 2017. Antimicrobials from human skin commensal bacteria protect against Staphylococcus aureus and are deficient in atopic dermatitis. *Science Translational Medicine*, 9(378), p.eaah4680.

Nayfach, S. et al., 2016. An integrated metagenomics pipeline for strain profiling reveals novel patterns of bacterial transmission and biogeography. *Genome Research*, 26(11), pp.1612–1625.

Niu, Q. et al., 2016. Changes in intestinal microflora of Caenorhabditis elegans following Bacillus nematocida B16 infection. *Nature Publishing Group*, pp.1–11.

Nyholm, S.V. & Graf, J., 2012. Knowing your friends: invertebrate innate immunity fosters beneficial bacterial symbioses. *Nature Publishing Group*, 10(12), pp.815–827.

Oliver, K.M., Moran, N.A. & Hunter, M.S., 2005. Variation in resistance to parasitism in aphids is due to symbionts not host genotype. *Proceedings of the National Academy of Sciences*, 102(36), pp.12795–12800.

Oliver, K.M., Smith, A.H. & Russell, J.A., 2013. Defensive symbiosis in the real world - advancing ecological studies of heritable, protective bacteria in aphids and beyond K. Clay, ed. *Functional Ecology*, 28(2), pp.341–355.

Olm, M.R. et al., 2017. Identical bacterial populations colonize premature infant gut, skin, and oral microbiomes and exhibit different in situ growth rates. *Genome Research*, 27(4), pp.601–612.

Ortiz, M.A. et al., 2014. A New Dataset of Spermatogenic vs. Oogenic Transcriptomes in the Nematode Caenorhabditis elegans. *G3: Genes, Genomes, Genetics*, 4(9), pp.1765–1772.

Page, A.P., Hamilton, A.J. & Maizels, R.M., 1992. Toxocara canis: Monoclonal antibodies to carbohydrate epitopes of secreted (TES) antigens localize to different secretion-related structures in infective larvae. *Experimental Parasitology*, 75(1), pp.56–71.

Papp, D., Csermely, P. & Sőti, C., 2012. A Role for SKN-1/Nrf in Pathogen Resistance and Immunosenescence in Caenorhabditis elegans F. M. Ausubel, ed. *PLoS Pathogens*, 8(4), pp.e1002673–11.

Park, J.-H. et al., 2016. Promotion of Intestinal Epithelial Cell Turnover by Commensal Bacteria: Role of Short-Chain Fatty Acids S. R. Singh, ed. *PLoS ONE*,

11(5), pp.e0156334–22.

Parker, B.J. et al., 2013. Symbiont-Mediated Protection against Fungal Pathogens in Pea Aphids: a Role for Pathogen Specificity? *Applied and Environmental Microbiology*, 79(7), pp.2455–2458.

Paskewitz, S.M., Li, B. & Kajla, M.K., 2008. Cloning and molecular characterization of two invertebrate-type lysozymes from Anopheles gambiae. *Insect Molecular Biology*, 17(3), pp.217–225.

Pees, B. et al., 2016. High Innate Immune Specificity through Diversified C-Type Lectin-Like Domain Proteins in Invertebrates. *Journal of Innate Immunity*, 8(2), pp.129–142.

Peleg, A.Y. et al., 2008. Prokaryote-eukaryote interactions identified by using Caenorhabditis elegans. *Proceedings of the National Academy of Sciences*, 105(38), pp.14585–14590.

Petersen, C., Dirksen, P. & Schulenburg, H., 2015. Why I need more ecology for genetic models such as C. elegans. *Trends in Genetics*, 31(3), pp.120–127.

Portal-Celhay, C. & Blaser, M.J., 2012. Competition and Resilience between Founder and Introduced Bacteria in the Caenorhabditis elegans Gut. *Infection and Immunity*, 80(3), pp.1288–1299.

Rubino, F. et al., 2017. Divergent functional isoforms drive niche specialisation for nutrient acquisition and use in rumen microbiome. 11(4), pp.932–944.

Samuel, B.S. et al., 2016. Caenorhabditis elegansresponses to bacteria from its natural habitats. *Proceedings of the National Academy of Sciences*, 113(27), pp.E3941–E3949.

Schulenburg, H. et al., 2008. Specificity of the innate immune system and diversity of C-type lectin domain (CTLD) proteins in the nematode Caenorhabditis elegans. *Immunobiology*, 213(3-4), pp.237–250.

Schulte, R.D. et al., 2011. Host-parasite local adaptation after experimental coevolution of Caenorhabditis elegans and its microparasite Bacillus thuringiensis. *Proceedings. Biological sciences / The Royal Society*, 278(1719), pp.2832–2839.

Schwarz, R.S., Moran, N.A. & Evans, J.D., 2016. Early gut colonizers shape parasite susceptibility and microbiota composition in honey bee workers. *Proceedings of the National Academy of Sciences of the United States of America*, 113(33), pp.9345–9350.

Schwarzer, M., Makki, K. & Storelli, G., 2016. Lactobacillus plantarum strain maintains growth of infant mice during chronic undernutrition. *Science*, 351(6257), pp.845–857.

Shapira, M. et al., 2006. A conserved role for a GATA transcription factor in regulating epithelial innate immune responses. *Proceedings of the National Academy of Sciences*, 103(38), pp.14086–14091.

Shin, S.C. et al., 2011. Drosophila microbiome modulates host developmental and metabolic homeostasis via insulin signaling. *Science*, 334(6056), pp.670–674.

Sorg, J.A. & Sonenshein, A.L., 2008. Bile Salts and Glycine as Cogerminants for Clostridium difficile Spores. *Journal of bacteriology*, 190(7), pp.2505–2512.

Spencer, W.C. et al., 2011. A spatial and temporal map of C. elegans gene expression. *Genome Research*, 21(2), pp.325–341.

Stecher, B. et al., 2012. Gut inflammation can boost horizontal gene transfer between pathogenic and commensal Enterobacteriaceae. *Proceedings of the National Academy of Sciences of the United States of America*, 109(4), pp.1269–1274.

Sulston, J.E. & Horvitz, H.R., 1977. Post-embryonic cell lineages of the nematode, Caenorhabditis elegans. *Developmental Biology*, 56(1), pp.110–156.

Taffoni, C. & Pujol, N., 2015. Mechanisms of innate immunity in C. elegansepidermis. *Tissue Barriers*, 3(4), pp.e1078432–8.

Troemel, E.R. et al., 2006. p38 MAPK Regulates Expression of Immune Response Genes and Contributes to Longevity in C. elegans. *PLoS Genetics*, 2(11), pp.e183–15.

van Baarlen, P. et al., 2011. Human mucosal in vivo transcriptome responses to three lactobacilli indicate how probiotics may modulate human cellular pathways. *Proceedings of the National Academy of Sciences*, 108(Supplement_1), pp.4562–4569.

van der Hoeven, R. et al., 2011. Ce-Duox1/BLI-3 Generated Reactive Oxygen Species Trigger Protective SKN-1 Activity via p38 MAPK Signaling during Infection in C. elegans F. M. Ausubel, ed. *PLoS Pathogens*, 7(12), p.e1002453.

Walker, T. et al., 2011. The wMel Wolbachia strain blocks dengue and invades caged Aedes aegypti populations. *Nature (News Feature)*, 476(7361), pp.450–453.

Walter, W., Sánchez-Cabo, F. & Ricote, M., 2015. GOplot: an R package for visually combining expression data with functional analysis. *Bioinformatics*, 31(17), pp.2912–2914.

Wheeler, J.M. & Thomas, J.H., 2006. Identification of a Novel Gene Family Involved in Osmotic Stress Response in Caenorhabditis elegans. *Genetics*, 174(3), pp.1327–1336.

Whitaker, W.R., Shepherd, E.S. & Sonnenburg, J.L., 2017. Tunable Expression Tools Enable Single-Cell Strain Distinction in the Gut Microbiome. *Cell*, 169(3),

pp.538–538.e12.

Wickham, H., 2009. *ggplot2: elegant graphics for data analysis*, New York, NY: Springer New York.

Wong, D., Bazopoulou, D., Pujol, N., Tavernarakis, N. & Ewbank, J.J., 2007a. Genome-wide investigation reveals pathogen-specific and shared signatures in the response of Caenorhabditis elegans to infection. *Genome Biology*, 8(9), pp.R194–18.

Wong, D., Bazopoulou, D., Pujol, N., Tavernarakis, N. & Ewbank, J.J., 2007b. Genome-wide investigation reveals pathogen-specific and shared signatures in the response of Caenorhabditis elegans to infection. *Genome Biology*, 8(9), pp.R194–18.

Supplementary Files

Supplementary file 1. R Markdown file outlining gut enumeration and protection analyses

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(ggplot2)
```

Define functions to plot equations on figures, taken from kdauria github code.

```
se <- function(x) sd(x)/sqrt(length(x))

stat_smooth_func <- function(mapping = NULL, data = NULL,
                         geom = "smooth", position = "identity",
                         ...,
                         method = "auto",
                         formula = y ~ x,
                         se = TRUE,
                         n = 80,
                         span = 0.75,
                         fullrange = FALSE,
                         level = 0.95,
                         method.args = list(),
                         na.rm = FALSE,
                         show.legend = NA,
                         inherit.aes = TRUE,
                         xpos = NULL,
                         ypos = NULL) {
  layer(
    data = data,
    mapping = mapping,
    stat = StatSmoothFunc,
    geom = geom,
    position = position,
    show.legend = show.legend,
    inherit.aes = inherit.aes,
    params = list(
      method = method,
      formula = formula,
      se = se,
      n = n,
      fullrange = fullrange,
      level = level,
      na.rm = na.rm,
```

```
      method.args = method.args,
      span = span,
      xpos = xpos,
      ypos = ypos,
      ...
    )
  )
}

StatSmoothFunc <- ggproto("StatSmooth", Stat,

                        setup_params = function(data, params) {
                          # Figure out what type of smoothing to do: loess for small data
sets,
                          # gam with a cubic regression basis for large data
                          # This is based on the size of the _largest_ group.
                          if (identical(params$method, "auto")) {
                            max_group <- max(table(data$group))

                            if (max_group < 1000) {
                              params$method <- "loess"
                            } else {
                              params$method <- "gam"
                              params$formula <- y ~ s(x, bs = "cs")
                            }
                          }
                          if (identical(params$method, "gam")) {
                            params$method <- mgcv::gam
                          }

                          params
                        },

                        compute_group = function(data, scales, method = "auto", formula =
 y~x,
                                                 se = TRUE, n = 80, span = 0.75, fullrang
e = FALSE,
                                                 xseq = NULL, level = 0.95, method.args =
 list(),
                                                 na.rm = FALSE, xpos=NULL, ypos=NULL) {
                          if (length(unique(data$x)) < 2) {
                            # Not enough data to perform fit
                            return(data.frame())
                          }

                          if (is.null(data$weight)) data$weight <- 1

                          if (is.null(xseq)) {
                            if (is.integer(data$x)) {
                              if (fullrange) {
                                xseq <- scales$x$dimension()
                              } else {
                                xseq <- sort(unique(data$x))
                              }
                            } else {
                              if (fullrange) {
                                range <- scales$x$dimension()
                              } else {
                                range <- range(data$x, na.rm = TRUE)
                              }
                              xseq <- seq(range[1], range[2], length.out = n)
                            }
                          }
```

```
                        # Special case span because it's the most commonly used model a
rgument
                        if (identical(method, "loess")) {
                          method.args$span <- span
                        }

                        if (is.character(method)) method <- match.fun(method)

                        base.args <- list(quote(formula), data = quote(data), weights =
 quote(weight))
                        model <- do.call(method, c(base.args, method.args))

                        m = model
                        eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(r
)^2~"="~r2,
                                         list(a = format(coef(m)[1], digits = 3),
                                              b = format(coef(m)[2], digits = 3),
                                              r2 = format(summary(m)$r.squared, digits
= 3)))
                        func_string = as.character(as.expression(eq))

                        if(is.null(xpos)) xpos = min(data$x)*0.9
                        if(is.null(ypos)) ypos = max(data$y)*0.9
                        data.frame(x=xpos, y=ypos, label=func_string)

                      },

                      required_aes = c("x", "y")
)

gac <- read.csv("~/Documents/King_Lab/Masters_thesis/gut_surv_data/cfus-7-5-17.csv")

gac.e <- subset(gac, treatment != "op50") %>% subset(treatment != "pm")

pairwise.t.test(log(gac.e$cfus), gac.e$treatment)

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  log(gac.e$cfus) and gac.e$treatment
##
##     ae     cce
## cce 0.0065 -
## se  0.4644 0.0025
##
## P value adjustment method: holm

# qqnorm(log(subset(gac, treatment == 'cce')$cfus)) qqnorm(log(subset(gac,
# treatment == 'se')$cfus)) qqnorm(log(subset(gac, treatment == 'ae')$cfus))

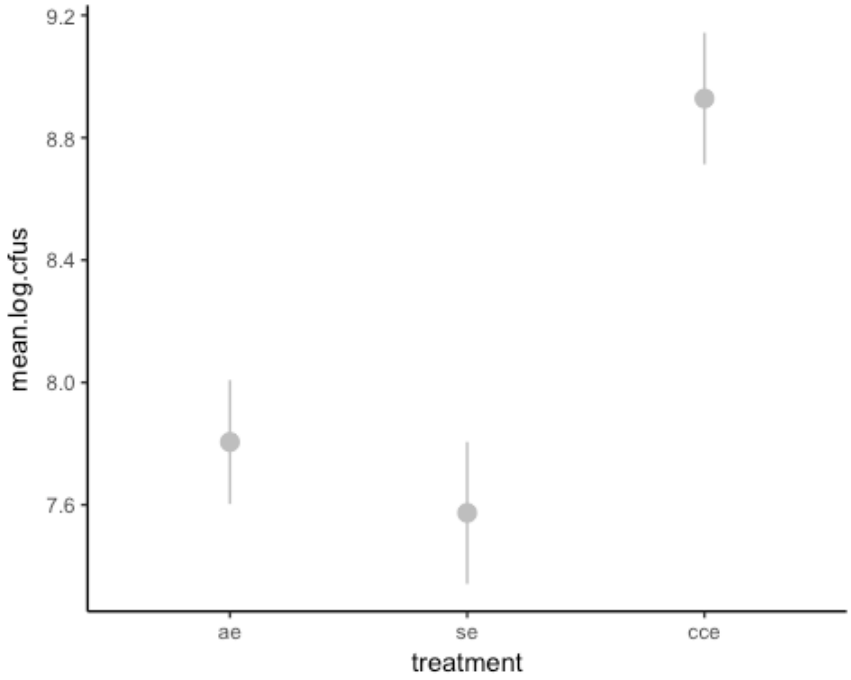
gac.e.m <- aggregate(data = gac.e, log(cfus) ~ treatment, mean)
colnames(gac.e.m)[2] <- "mean.log.cfus"
se <- function(x) sd(x)/sqrt(length(x))
gac.e.se <- aggregate(data = gac.e, log(cfus) ~ treatment, se)
colnames(gac.e.se)[2] <- "se.log.cfus"

gac.e.m.se <- cbind(gac.e.m, gac.e.se)

cfu_limits <- aes(ymax = gac.e.m.se$mean.log.cfus + gac.e.m.se$se.log.cfus,
    ymin = gac.e.m.se$mean.log.cfus - gac.e.m.se$se.log.cfus)

p <- ggplot(gac.e.m.se, aes(treatment, mean.log.cfus))
```

```
p + geom_point(size = 5, shape = 20, color = "grey") + theme_classic() + scale_x_discre
te(limits = c("ae",
    "se", "cce")) + geom_pointrange(cfu_limits, color = "grey")
```




```
# Make without log for summary stats
gac.e.m <- aggregate(data = gac.e, (cfus) ~ treatment, mean)
colnames(gac.e.m)[2] <- "mean.log.cfus"
se <- function(x) sd(x)/sqrt(length(x))
gac.e.se <- aggregate(data = gac.e, (cfus) ~ treatment, se)
colnames(gac.e.se)[2] <- "se.log.cfus"

gac.e.m.se <- cbind(gac.e.m, gac.e.se)
gac.e.m.se

##   treatment mean.log.cfus treatment se.log.cfus
## 1        ae      2663.556        ae    542.6499
## 2       cce      8201.468       cce   1539.6692
## 3        se      2125.407        se    365.1053

sv <- read.csv("~/Documents/King_Lab/Masters_thesis/gut_surv_data/surv-7-5-17.csv")

sv.e <- subset(sv, treatment != "op50")

pairwise.wilcox.test(log(sv.e$prop.dead), sv.e$treatment)

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

## Warning in wilcox.test.default(xi, xj, paired = paired, ...): cannot
## compute exact p-value with ties

##
##  Pairwise comparisons using Wilcoxon rank sum test
##
## data:  log(sv.e$prop.dead) and sv.e$treatment
##
##     ae    cce
```

```
## cce 0.048 -
## se  0.142 0.048
##
## P value adjustment method: holm

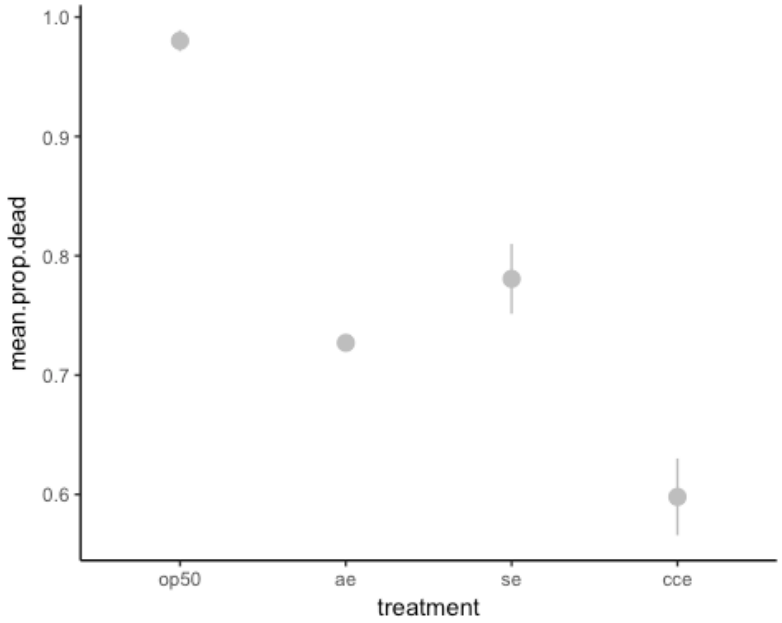
sv.e.m <- aggregate(data = sv, prop.dead ~ treatment, mean)
colnames(sv.e.m)[2] <- "mean.prop.dead"
se <- function(x) sd(x)/sqrt(length(x))
sv.e.se <- aggregate(data = sv, prop.dead ~ treatment, se)
colnames(sv.e.se)[2] <- "se.prop.dead"

sv.e.m.se <- merge(sv.e.m, sv.e.se)


surv_limits <- aes(ymax = sv.e.m.se$mean.prop.dead + sv.e.m.se$se.prop.dead,
    ymin = sv.e.m.se$mean.prop.dead - sv.e.m.se$se.prop.dead)


p2 <- ggplot(sv.e.m.se, aes(treatment, mean.prop.dead))

p2 + geom_point(size = 5, shape = 20, color = "grey") + theme_classic() + scale_x_discr
ete(limits = c("op50",
    "ae", "se", "cce")) + geom_pointrange(surv_limits, color = "grey")
```



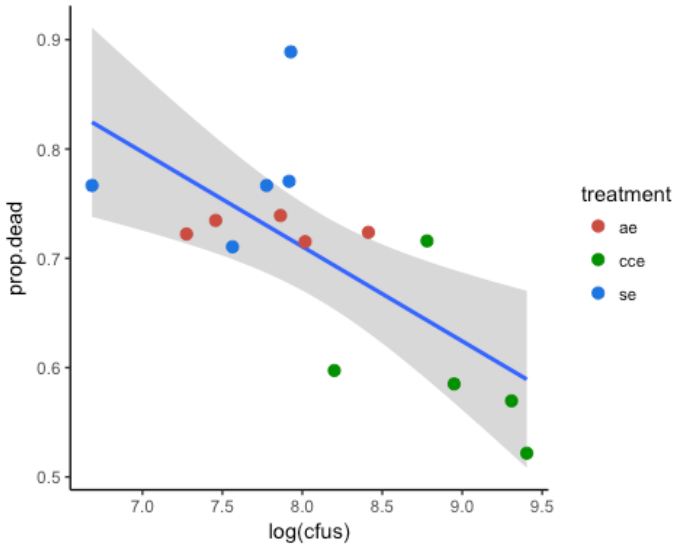
Plotting CFUs against protection

```
gac.e.o <- subset(gac, treatment != "pm")

sv.gac <- merge(gac.e.o, sv, by = c("treatment", "rep"))
sv.gac.ef <- subset(sv.gac, treatment != "op50")

p3 <- ggplot(sv.gac.ef, aes(log(cfus), prop.dead)) + scale_colour_hue(l = 50) +
    geom_smooth(method = lm, se = TRUE, fullrange = FALSE) + geom_point(aes(color = tre
atment),
    shape = 20, size = 4) + theme_classic()
# stat_smooth_func(geom='text',method='lm',hjust=0,parse=TRUE) +
# theme_classic()

p3
```




```
# prop dead as response and cfus as predictor
cor.test(sv.gac.ef$prop.dead, sv.gac.ef$cfus, method = "p")
```

```
##
##  Pearson's product-moment correlation
##
## data:  sv.gac.ef$prop.dead and sv.gac.ef$cfus
## t = -4.4153, df = 13, p-value = 0.0006977
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9212782 -0.4348218
## sample estimates:
##       cor
## -0.7745574
```

Save objects for future correlations.

```
write.csv(sv.gac, file = "~/Documents/King_Lab/Masters_thesis/gut_surv_data/colonize_surv.csv")

devtools::session_info()

## Session info -------------------------------------------------------------

##  setting  value
##  version  R version 3.4.0 (2017-04-21)
##  system   x86_64, darwin15.6.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/Los_Angeles
##  date     2017-06-01

## Packages -----------------------------------------------------------------

##  package    * version date       source
##  assertthat   0.2.0   2017-04-11 cran (@0.2.0)
##  backports    1.0.5   2017-01-18 CRAN (R 3.4.0)
##  base       * 3.4.0   2017-04-21 local
##  codetools    0.2-15  2016-10-05 CRAN (R 3.4.0)
##  colorspace   1.3-2   2016-12-14 CRAN (R 3.4.0)
##  compiler     3.4.0   2017-04-21 local
##  datasets   * 3.4.0   2017-04-21 local
```

```
##  DBI          0.6-1   2017-04-01 CRAN (R 3.4.0)
##  devtools     1.13.1  2017-05-13 CRAN (R 3.4.0)
##  digest       0.6.12  2017-01-27 CRAN (R 3.4.0)
##  dplyr      * 0.5.0   2016-06-24 cran (@0.5.0)
##  evaluate     0.10    2016-10-11 CRAN (R 3.4.0)
##  formatR      1.5     2017-04-25 CRAN (R 3.4.0)
##  ggplot2    * 2.2.1   2016-12-30 CRAN (R 3.4.0)
##  graphics   * 3.4.0   2017-04-21 local
##  grDevices  * 3.4.0   2017-04-21 local
##  grid         3.4.0   2017-04-21 local
##  gtable       0.2.0   2016-02-26 CRAN (R 3.4.0)
##  htmltools    0.3.6   2017-04-28 CRAN (R 3.4.0)
##  knitr        1.15.1  2016-11-22 CRAN (R 3.4.0)
##  labeling     0.3     2014-08-23 CRAN (R 3.4.0)
##  lazyeval     0.2.0   2016-06-12 CRAN (R 3.4.0)
##  magrittr     1.5     2014-11-22 CRAN (R 3.4.0)
##  memoise      1.1.0   2017-04-21 CRAN (R 3.4.0)
##  methods    * 3.4.0   2017-04-21 local
##  munsell      0.4.3   2016-02-13 CRAN (R 3.4.0)
##  plyr         1.8.4   2016-06-08 CRAN (R 3.4.0)
##  R6           2.2.1   2017-05-10 CRAN (R 3.4.0)
##  Rcpp         0.12.10 2017-03-19 CRAN (R 3.4.0)
##  rmarkdown    1.5     2017-04-26 CRAN (R 3.4.0)
##  rprojroot    1.2     2017-01-16 CRAN (R 3.4.0)
##  scales       0.4.1   2016-11-09 CRAN (R 3.4.0)
##  stats      * 3.4.0   2017-04-21 local
##  stringi      1.1.5   2017-04-07 CRAN (R 3.4.0)
##  stringr      1.2.0   2017-02-18 CRAN (R 3.4.0)
##  tibble       1.3.0   2017-04-01 CRAN (R 3.4.0)
##  tools        3.4.0   2017-04-21 local
##  utils      * 3.4.0   2017-04-21 local
##  withr        1.0.2   2016-06-20 CRAN (R 3.4.0)
##  xtable     * 1.8-2   2016-02-05 CRAN (R 3.4.0)
```

Supplementary file 2. Snakemake commands for processing RNA reads with Trimmomatic and kallisto.

```
N_THREADS = 16

ANNO_PRE = "annotation/cele_trans"
ANNO_FA = ANNO_PRE + ".fa.gz"
KAL_IDX = ANNO_PRE + ".kidx"

SAMPLES = ['WTCHG_339857_211126', 'WTCHG_339857_212138', 'WTCHG_339857_213150',
 'WTCHG_339857_214162', 'WTCHG_339857_215174', 'WTCHG_339857_216186', 'WTCHG_339857_217103',
 'WTCHG_339857_218115','WTCHG_339857_219127', 'WTCHG_339857_220139','WTCHG_339857_221151',

'WTCHG_339857_222163','WTCHG_339857_223175','WTCHG_339857_224187','WTCHG_339857_225104','WTCHG_339
857_226116']

rule all:
    input:
        expand('results/paired/{id}/kallisto/abundance.h5', id = SAMPLES)

rule trimmomatic_paired:
    input:
        forward = "data/{id}_1.fastq.gz",
        reverse = "data/{id}_2.fastq.gz",
    output:
        forward_paired = "data/trimmed/{id}_paired_1.fastq.gz",
        forward_unpaired = "data/trimmed/{id}_unpaired_1.fastq.gz",
        reverse_paired = "data/trimmed/{id}_paired_2.fastq.gz",
        reverse_unpaired = "data/trimmed/{id}_unpaired_2.fastq.gz"
    message: "Trimming and filtering {input.forward} and {input.reverse}"
```

```
    shell:
        """
        java -jar /home/share/software/Trimmomatic-0.32/trimmomatic-0.32.jar PE {input.forward}
{input.reverse} {output.forward_paired} \
        {output.forward_unpaired} {output.reverse_paired} {output.reverse_unpaired} \
        ILLUMINACLIP:/home/share/software/Trimmomatic-0.32/adapters/TruSeq3-PE.fa:2:30:10
LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:50
        """

rule kallisto_paired:
    input:
        'data/trimmed/{id}_paired_1.fastq.gz',
        'data/trimmed/{id}_paired_2.fastq.gz',
        KAL_IDX
    output:
        'results/paired/{id}/kallisto',
        'results/paired/{id}/kallisto/abundance.h5'
    threads: N_THREADS
    shell:
        'kallisto quant '
        '-i {KAL_IDX} '
        '-b 30 '
        '--bias '
        '-t {threads} '
        '-o {output[0]} '
        '{input[0]} {input[1]}'

rule get_annotation:
    output:
        ANNO_FA
    shell:
        'wget -O {output} '

'http://bio.math.berkeley.edu/kallisto/transcriptomes/Caenorhabditis_elegans.WBcel235.rel79.cdna.a
ll.fa.gz'

rule kallisto_index:
    input:
        ANNO_FA
    output:
        KAL_IDX
    shell:
        'kallisto index '
        '-i {output} {input}'
```

Supplementary file 3. R Markdown file outlining differential expression and GO term analysis.

```
Load libraries
.bioc_packages <- c("devtools", "sleuth", "biomaRt", "VennDiagram")
.cran_packages <- c("GOplot", "ggplot2", "plyr", "dplyr")

.inst <- .cran_packages %in% installed.packages()
if (any(!.inst)) {
    install.packages(.cran_packages[!.inst])
}
.inst <- .bioc_packages %in% installed.packages()
if (any(!.inst)) {
    source("http://bioconductor.org/biocLite.R")
    biocLite(.bioc_packages[!.inst], ask = F)
}
```

```
library(sleuth)
sapply(c(.cran_packages, .bioc_packages), require, character.only = TRUE)
```

```
##      GOplot     ggplot2        plyr       dplyr    devtools      sleuth
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##     biomaRt VennDiagram
##        TRUE        TRUE
```

```
set.seed(100)
```

Input results and sample data

Set base directory for results and load sample names

```
base_dir <- "~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/"

s2c <- read.csv("/Users/Dylan/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/sampnames
.csv",
    header = TRUE, stringsAsFactors = FALSE)

sample_id <- dir(file.path(base_dir, "results", "paired"))
```

Set kallisto directory, add file paths to data, and make data subsets for comparisons

```
kal_dirs <- sapply(sample_id, function(id) file.path(base_dir, "results", "paired",
    id, "kallisto"))

s2c <- mutate(s2c, path = kal_dirs)

# Make object for comparing treatments to ancestor
s2cno <- s2c[!(s2c$condition == "op50"), ]
# evolved ef (SE) to ancestor
soSE <- s2cno[!(s2cno$condition == "CCE"), ]
# cocolonized evolved (CCE) to ancestor
soCCE <- s2cno[!(s2cno$condition == "SE"), ]
# CCE to SE
soCCESE <- s2cno[!(s2cno$condition == "AE"), ]
# CCE to OP50
soCCEOP <- s2c[!(s2c$condition == "SE"), ]
soCCEOP <- soCCEOP[!(s2c$condition == "AE"), ] %>% na.omit()
# SE to OP50
soSEOP <- s2c[!(s2c$condition == "CCE"), ]
soSEOP <- soSEOP[!(s2c$condition == "AE"), ] %>% na.omit()
# Make object comparing ancestor to op50
s2cpp <- s2c[!(s2c$condition == "SE"), ]
s2cpp <- s2cpp[!(s2cpp$condition == "CCE"), ]
```

Get gene names from ensembl

Use biomaRt package to pull gene names and other info (e.g., wormbase IDs) from ensembl

```
# access ensemble datasets
ensembl87 = useEnsembl(biomart = "ensembl", version = 87)
```

```
## Note: requested host was redirected from e87.ensembl.org to http://dec2016.archive.e
nsembl.org:80/biomart/martservice
## When using archived Ensembl versions this sometimes can result in connecting to a ne
wer version than the intended Ensembl version
## Check your ensembl version using listMarts(mart)
```

```
mart <- biomaRt::useMart(biomart = "ENSEMBL_MART_ENSEMBL", dataset = "celegans_gene_ens
embl",
    host = "www.ensembl.org")

# Add transcript id, ensemble gene id (WormBase here) and external gene name
# (e.g., asp-3)
t2g <- biomaRt::getBM(attributes = c("ensembl_transcript_id", "ensembl_gene_id",
    "external_gene_name", "description"), mart = mart)

# Can add these go descriptors other information, all of which is seen at:
# listAttributes(mart)

t2g <- dplyr::rename(t2g, target_id = ensembl_transcript_id, ens_gene = ensembl_gene_id
,
    ext_gene = external_gene_name)
```

Construct sleuth objects, fit model and run Wald Test

First I compare two evolved treatments (SE and CCE) to ancestor (AE)

```
# construct sleuth object
soSE <- sleuth_prep(soSE, ~condition, target_mapping = t2g)
# fit the model
soSE <- sleuth_fit(soSE)
# run wald test, with beta (comparing to control) a SE
soSE <- sleuth_wt(soSE, which_beta = "conditionSE")

# same but for CCE
soCCE <- sleuth_prep(soCCE, ~condition, target_mapping = t2g)
soCCE <- sleuth_fit(soCCE)
soCCE <- sleuth_wt(soCCE, which_beta = "conditionCCE")

# Same again but using all w E. faecalis for visualization purpses, can see
# with sleuth_live
so <- sleuth_prep(s2cno, ~condition, target_mapping = t2g)
so <- sleuth_fit(so)
so <- sleuth_wt(so, which_beta = "conditionCCE")
# Write a dataframe of normalized tpm values for correlations in other
# analyses
somx <- sleuth_to_matrix(so, "obs_norm", "tpm")
somx <- t(somx$data)
somx <- data.frame(somx)
somxsd <- s2cno
row.names(somxsd) <- somxsd$sample
somxdf <- merge(somxsd, somx, by = 0, all = TRUE)
write.csv(somxdf, "~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/somxdf.csv")
```

repeat treatment across sample IDs merge by treatment

Then directly compare CCE to SE

```
# Baseline is automatically set alphabetically, so in order to set baseline
# to compare CCE to SE I need to reset it
cond <- factor(soCCESE$condition)
cond <- relevel(cond, ref = "SE")
md <- model.matrix(~cond, soCCESE)

soCCESE <- sleuth_prep(soCCESE, md, target_mapping = t2g)
soCCESE <- sleuth_fit(soCCESE)
soCCESE <- sleuth_wt(soCCESE, which_beta = "condCCE")
```

Compare CCE to OP50

```
# Baseline is automatically set alphabetically, so in order to set baseline
# to compare CCE to op50 I need to reset it
cond2 <- factor(soCCEOP$condition)
cond2 <- relevel(cond2, ref = "op50")
md2 <- model.matrix(~cond2, soCCEOP)

soCCEOP <- sleuth_prep(soCCEOP, md2, target_mapping = t2g)
soCCEOP <- sleuth_fit(soCCEOP)
soCCEOP <- sleuth_wt(soCCEOP, which_beta = "cond2CCE")
```

Compare SE to OP50

```
# Baseline is automatically set alphabetically, so in order to set baseline
# to compare CCE to op50 I need to reset it
cond3 <- factor(soSEOP$condition)
cond3 <- relevel(cond3, ref = "op50")
md3 <- model.matrix(~cond3, soSEOP)

soSEOP <- sleuth_prep(soSEOP, md3, target_mapping = t2g)
soSEOP <- sleuth_fit(soSEOP)
soSEOP <- sleuth_wt(soSEOP, which_beta = "cond3SE")
```

Compare AE to OP50

```
# Baseline is automatically set alphabetically, so in order to set baseline
# to compare ancestor e faecalis to op50 I need to reset it
cond4 <- factor(s2cpp$condition)
cond4 <- relevel(cond4, ref = "op50")
md4 <- model.matrix(~cond4, s2cpp)

sopp <- sleuth_prep(s2cpp, md4, target_mapping = t2g)
sopp <- sleuth_fit(sopp)
sopp <- sleuth_wt(sopp, which_beta = "cond4AE")
```

Extract results from Wald test results from all comparisons

```
# SE compared to ancestor
soSE.res <- sleuth_results(soSE, "conditionSE", test_type = "wt")
# Order by adj-p, here called qval
soSE.res <- soSE.res[order(soSE.res$qval), ]
# Subset to significant (q < 0.05)
soSE.res.sig <- subset(soSE.res, qval <= 0.05)
# write.csv(soSE.res.sig,file = 'Ee.res.sig.csv')


# CCE compared to ancestor
soCCE.res <- sleuth_results(soCCE, "conditionCCE", test_type = "wt")
soCCE.res <- soCCE.res[order(soCCE.res$qval), ]
soCCE.res.sig <- subset(soCCE.res, qval <= 0.05)
# write.csv(soCCE.res.sig,file = 'EeS.res.sig.csv')


# CCE to SE
ccese.res <- sleuth_results(soCCESE, "condCCE", test_type = "wt")
ccese.res <- ccese.res[order(ccese.res$qval), ]
ccese.res.sig <- subset(ccese.res, qval <= 0.05)
# write.csv(ccese.res.sig,file = 'ccese.res.sig.csv')

# CCE to OP50
cceop.res <- sleuth_results(soCCEOP, "cond2CCE", test_type = "wt")
cceop.res <- cceop.res[order(cceop.res$qval), ]
cceop.res.sig <- subset(cceop.res, qval <= 0.05)
```

```
# SE to OP50
seop.res <- sleuth_results(soSEOP, "cond3SE", test_type = "wt")
seop.res <- seop.res[order(seop.res$qval), ]
seop.res.sig <- subset(seop.res, qval <= 0.05)

# Same comparing ancestor to OP50
sopp.res <- sleuth_results(sopp, "cond4AE", test_type = "wt")
sopp.res <- sopp.res[order(sopp.res$qval), ]
sopp.res.sig <- subset(sopp.res, qval <= 0.05)
```

Comparing SE and CCE to ancestor

Summarizing DEGs in set SE, set CCE and intersection

How many DEGs in conditions? SE:

```
length(unique(soSE.res.sig$target_id))

## [1] 135
```

CCE:

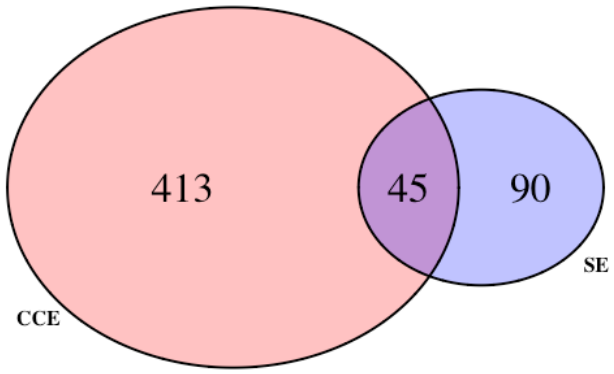
```
length(unique(soCCE.res.sig$target_id))

## [1] 458
```

Plot Venn diagram to see # of overlapping genes

```
venn.plot <- venn.diagram(list(soSE.res.sig$target_id, soCCE.res.sig$target_id),
    NULL, fill = c("blue", "red"), alpha = c(0.3, 0.3), cex = 2, cat.fontface = 2,
    category.names = c("SE", "CCE"))

grid.draw(venn.plot)
```



Plot differentially expressed transcripts by gene_id.

First plot the intersection

```
# Rep condition names
soSE.res.sig[, "condition"] <- rep("SE", length(rownames(soSE.res.sig)))
soCCE.res.sig[, "condition"] <- rep("CCE", length(rownames(soCCE.res.sig)))

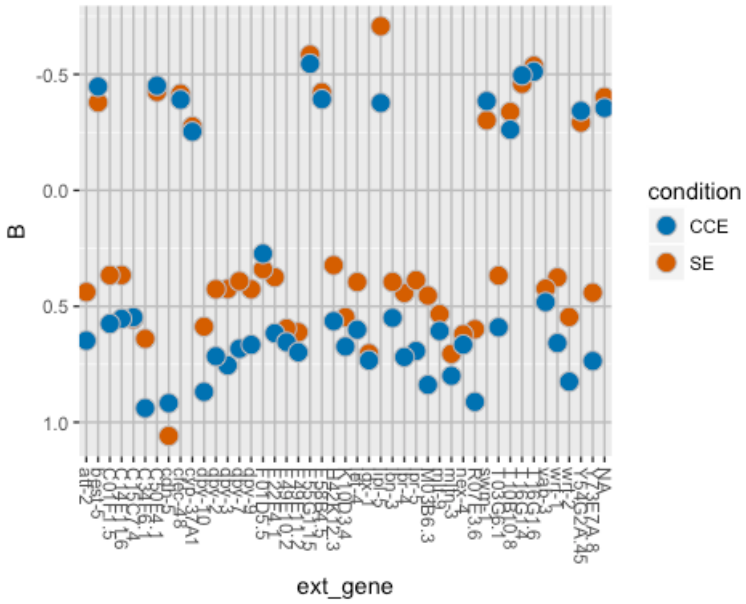
# Bind
res.sig <- rbind(soSE.res.sig, soCCE.res.sig)
```

```
# Intersection of targets between two
res.inter <- intersect(soSE.res.sig$target_id, soCCE.res.sig$target_id)
# Subset table of significant transcripts that intersect
res.inter.sig <- subset(res.sig, target_id %in% res.inter)
# write csv
# write.csv(res.inter.sig,'~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/res_in
ter_sig.csv')

# Make limits for error bars
res.inter.sig.limits <- aes(ymax = res.inter.sig$b + res.inter.sig$se_b, ymin = res.int
er.sig$b -
    res.inter.sig$se_b)

# plot
de.inter = ggplot(res.inter.sig, aes(x = ext_gene, y = b, color = condition,
    fill = condition)) + geom_point(shape = 21, size = 4, color = "grey") +
    theme(panel.grid.major = element_line(colour = "grey"), axis.text.x = element_text(
angle = -90,
        hjust = 0, vjust = 0.5)) + ylab("B") + scale_fill_manual(values = c("#0072B2",
    "#D55E00")) + scale_y_reverse()

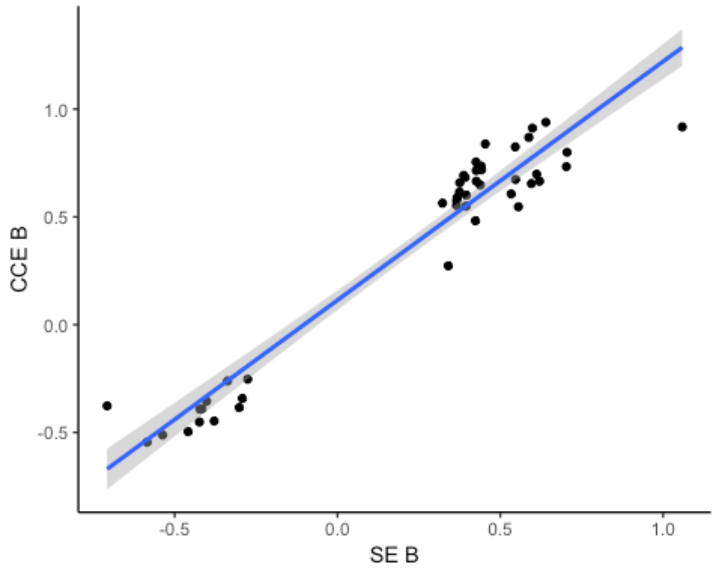
de.inter
```




```
res.inter.sig_cce <- subset(res.inter.sig, condition == "CCE")
colnames(res.inter.sig_cce)[4] <- "cceB"
res.inter.sig_se <- subset(res.inter.sig, condition == "SE")
colnames(res.inter.sig_se)[4] <- "seB"

res.inter.sig.merged <- merge(res.inter.sig_cce, res.inter.sig_se, by = "target_id")

p <- ggplot(res.inter.sig.merged, aes(seB, cceB))
# by default includes 95% confidence region
p + geom_point() + geom_smooth(method = "lm") + theme_classic() + xlab("SE B") +
    ylab("CCE B")
```




```
cor.test(res.inter.sig.merged$seB, res.inter.sig.merged$cceB, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  res.inter.sig.merged$seB and res.inter.sig.merged$cceB
## t = 24.514, df = 43, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9386746 0.9813055
## sample estimates:
##       cor
## 0.9660343
```

```

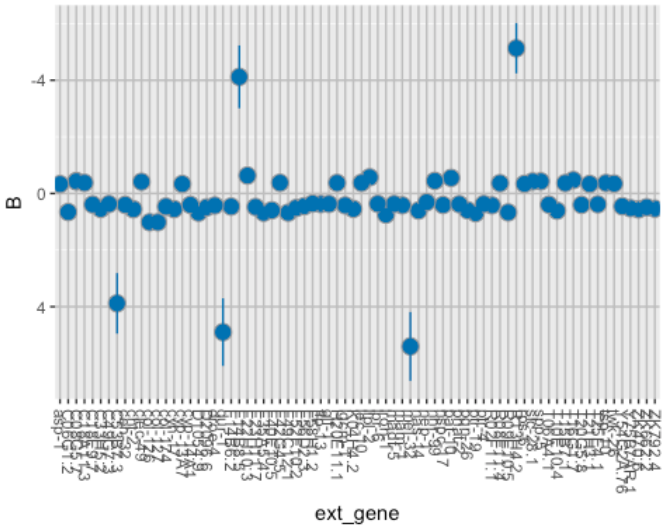
Next plot top 75 found just in set SE

```
# Now make an object that are found in only set SE by removing those found
# in set CCE from the dataframe (also removes intersection)
soSE.res.sig.set <- subset(soSE.res.sig, !(target_id %in% soCCE.res.sig$target_id))
# remove NA for plotting
soSE.res.sig.set <- soSE.res.sig.set[complete.cases(soSE.res.sig.set),]
# Plot top 75
soSE.res.sig.set.top <- soSE.res.sig.set[order(-abs(soSE.res.sig.set$b)),][1:75,
    ]

soSElimits <- aes(ymax = soSE.res.sig.set.top$b + soSE.res.sig.set.top$se_b,
    ymin = soSE.res.sig.set.top$b - soSE.res.sig.set.top$se_b)

de.soSE.res.sig.set = ggplot(soSE.res.sig.set.top, aes(x = ext_gene, y = b,
    color = condition, fill = condition)) + geom_point(colour = "#999999", fill = "#007
2B2",
    shape = 21, size = 4) + theme(panel.grid.major = element_line(colour = "grey"),
    axis.text.x = element_text(angle = -90, hjust = 0, vjust = 0.5)) + ylab("B") +
    scale_y_reverse() + geom_pointrange(soSElimits, fill = "#0072B2", color = "#0072B2"
)

de.soSE.res.sig.set
```
```



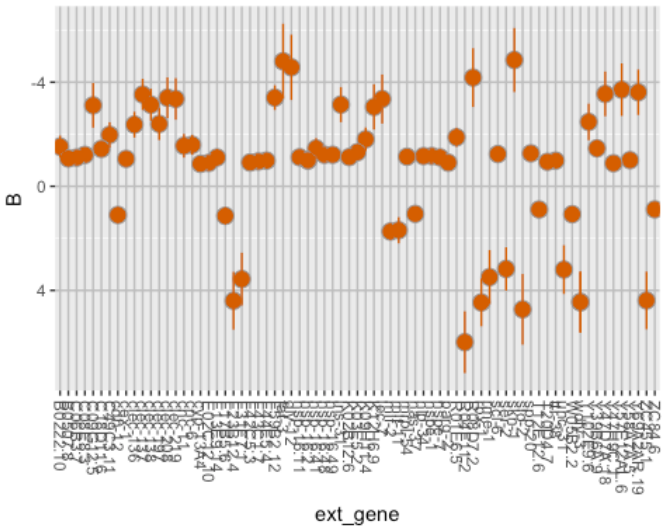
Then plot top 75 from CCE set

```
# same for CCE
soCCE.res.sig.set <- subset(soCCE.res.sig, !(target_id %in% soSE.res.sig$target_id))
soCCE.res.sig.set <- soCCE.res.sig.set[complete.cases(soCCE.res.sig.set), ]
soCCE.res.sig.set.top <- soCCE.res.sig.set[order(-abs(soCCE.res.sig.set$b)),
    ][1:75, ]

soCCElimits <- aes(ymax = soCCE.res.sig.set.top$b + soCCE.res.sig.set.top$se_b,
    ymin = soCCE.res.sig.set.top$b - soCCE.res.sig.set.top$se_b)

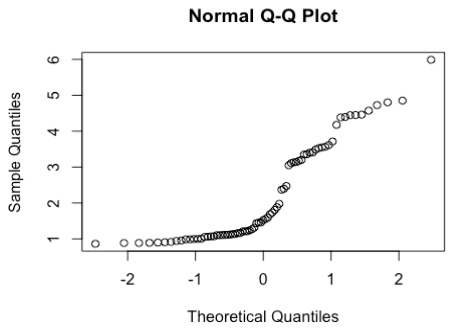
de.soCCE.res.sig.set = ggplot(soCCE.res.sig.set.top, aes(x = ext_gene, y = b)) +
    geom_point(colour = "#999999", fill = "#D55E00", shape = 21, size = 4) +
    theme(panel.grid.major = element_line(colour = "grey"), axis.text.x = element_text(
angle = -90,
        hjust = 0, vjust = 0.5)) + ylab("B") + scale_y_reverse() + geom_pointrange(soCC
Elimits,
    fill = "#D55E00", colour = "#D55E00")


de.soCCE.res.sig.set
```

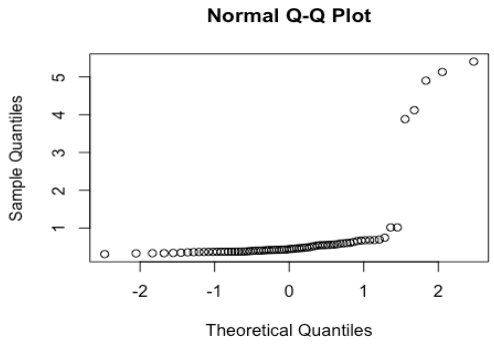


Test to see if absolute value B from top DEGs in CCE treatment are larger than those from SE treatment.

```
# check distriubtion
qqnorm(abs(soCCE.res.sig.set.top$b))
```




```
qqnorm(abs(soSE.res.sig.set.top$b))
```




```
# Nope, use Mann-Whitney
wilcox.test(abs(soCCE.res.sig.set.top$b), abs(soSE.res.sig.set.top$b), "greater")

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  abs(soCCE.res.sig.set.top$b) and abs(soSE.res.sig.set.top$b)
## W = 5247, p-value < 2.2e-16
## alternative hypothesis: true location shift is greater than 0
```

Yes, they are significantly different. Now report means and standard error for descriptive stats.

```
# yes, sig different now report means
mean(abs(soSE.res.sig.set.top$b))

## [1] 0.7616914

mean(abs(soCCE.res.sig.set.top$b))

## [1] 2.219368

se <- function(x) sd(x)/sqrt(length(x))
se(abs(soSE.res.sig.set.top$b))

## [1] 0.1242481
```

```
se(abs(soCCE.res.sig.set.top$b))

## [1] 0.1585339
```

DAVID functional enrichment analysis

Now write gene table (with ext_gene) and open DAVID to run functional annotation analysis. In DAVID I use a gene enrichment analysis. Go to DAVID and input a list of significantly diff expressed genes (P < 0.05) using official_gene_symbol which is ext_gene by sleuth/biomart. Here I use DAVID 6.8, which after a lot of controversy is the much awaited rebuild.

Export data for use in DAVID

```
# Write unique gene names for SE to ancestor comparison
soSE.res.sig.unique.extgene <- unique(soSE.res.sig$ext_gene)
write.table(soSE.res.sig.unique.extgene, "~/Desktop/soSE.res.sig.unique.extgene.csv",
    row.names = FALSE, col.names = FALSE)

# Write unique gene names for CCE to ancestor comparison
soCCE.res.sig.unique.extgene <- unique(soCCE.res.sig$ext_gene)
write.table(soCCE.res.sig.unique.extgene, "~/Desktop/soCCE.res.sig.unique.extgene.csv",

    row.names = FALSE, col.names = FALSE)

# Write unique gene names for CCE to SE comparison
ccese.res.sig.unique.extgene <- unique(ccese.res.sig$ext_gene)
write.table(ccese.res.sig.unique.extgene, "~/Documents/King_Lab/Masters_thesis/RNASeq/s
leuth/csv/ccese.res.sig.unique.extgene.csv",
    row.names = FALSE, col.names = FALSE)
```

Use GOPlot to plot chords of encriched GO terms and DEGs

Using the GOPlot library I can plot our DAVID output. First I write a function for converting our data to one that fits for GOplot.

```
# Write a function to input DAVID functional annotation output and convert
# to columns that are taken in by GOPlot. Also only keep terms functionally
# expressed at P<0.05
GOplotdatDavid <- function(soSE.res.sig.david) {
    Category <- soSE.res.sig.david$Category
    ID <- soSE.res.sig.david$Term
    # Delete everything after character for ID
    ID <- gsub("~.*", "", ID)
    Term <- soSE.res.sig.david$Term
    # Delete everything before character for term
    Term <- gsub(".*~", "", Term)
    Genes <- soSE.res.sig.david$Genes
    adj_pval <- soSE.res.sig.david$Benjamini
    gopdat <- data.frame(Category, ID, Term, Genes, adj_pval)
    gopdat <- subset(gopdat, adj_pval <= 0.05)
    return(gopdat)
}

# Take in sleuth object and output genelist for GOPlot
GOplotdatGeneList <- function(soSE.res.sig) {
    # repeats when I have columns w/ GO terms, only keep others and pull
    # uniques Should not do this since multiple transcripts can match to a
    # ext_gene, need to just remove go term
    ID <- soSE.res.sig$ext_gene
    B <- soSE.res.sig$b
    # GOPlot calls for LogFC but I calculated Beta value from Wald-Test, so use
    # this in LogFC column and change in figure
```

```
    logFC <- soSE.res.sig$b
    adj.P.Val <- soSE.res.sig$qval
    gopdat <- data.frame(ID, B, logFC, adj.P.Val)
    return(gopdat)
}
```

Read DAVID output and make figs

For SE to ancestor comparison, prepare DAVID data and plot circos figure

```
# Import DAVID output and run function
soSE.res.sig.f.david <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv
/soSE.res.sig.func.annotation.csv")
soSE.res.sig.david <- GOplotdatDavid(soSE.res.sig.f.david)

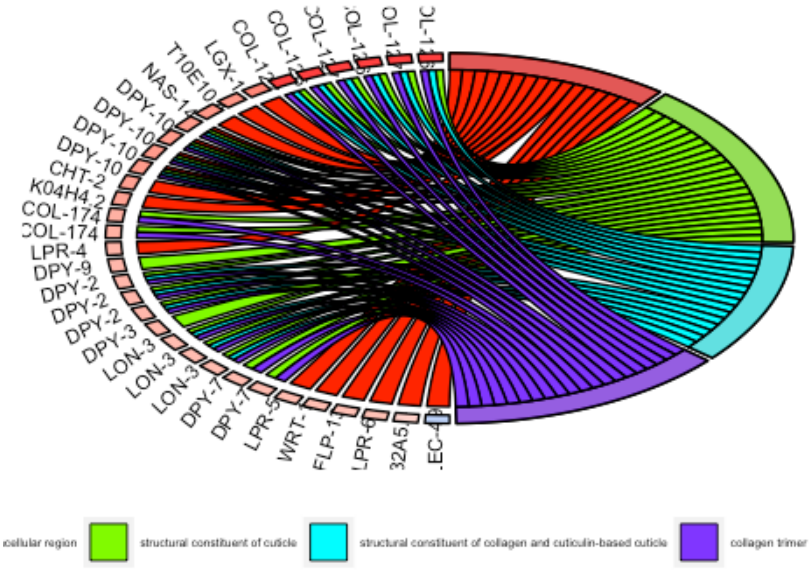
# Make gene list
soSE.res.sig.genelist <- GOplotdatGeneList(soSE.res.sig)
# Can also subset a list of processes to plot soSE.process <-
# soSE.res.sig.david$Term Make data for GOPlot
circ <- circle_dat(soSE.res.sig.david, soSE.res.sig.genelist)

# Make list of genes for chord, this just uses all since w/ Evolved E.
# faecalis I have few
genes <- circ$genes
logFC <- circ$logFC
soSE.res.sig.genes <- data.frame(genes, logFC)

# Make data for chord figure
chord <- chord_dat(data = circ, genes = soSE.res.sig.genes)
goChord <- GOChord(chord, space = 0.02, gene.order = "logFC", gene.space = 0.25,
    gene.size = 3, process.label = 5, lfc.min = -3, lfc.max = 3, nlfc = 1)
goChord

## Warning: Using size for a discrete variable is not advised.

## Warning: Removed 4 rows containing missing values (geom_point).
```



Same for CCE to ancestor

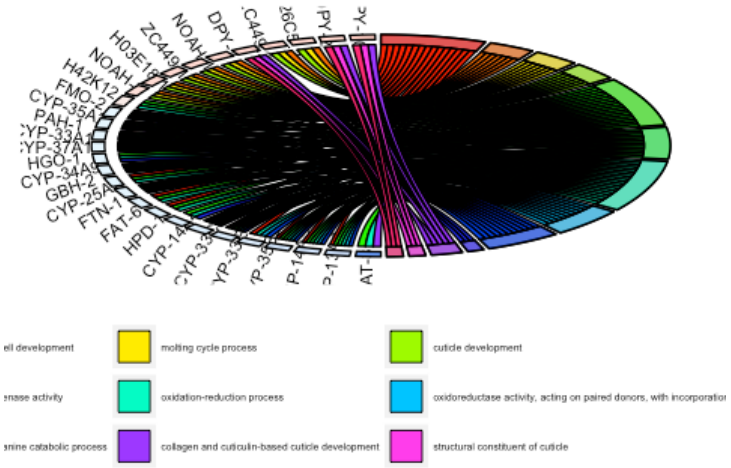
```
# Same but with S. aureus evolved E. faecalis
soCCE.res.sig.f.david <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/soCCE.res.sig.func.annotation.csv")
soCCE.res.sig.david <- GOplotdatDavid(soCCE.res.sig.f.david)
soCCE.res.sig.genelist <- GOplotdatGeneList(soCCE.res.sig)
soCCE.process <- soCCE.res.sig.david$Term
circSA <- circle_dat(soCCE.res.sig.david, soCCE.res.sig.genelist)
genes <- circSA$genes
logFC <- circSA$logFC
soCCE.res.sig.genes <- data.frame(genes, logFC)
soCCE.res.sig.genes <- unique(soCCE.res.sig.genes)
soCCE.res.sig.genes <- soCCE.res.sig.genes[order(-abs(soCCE.res.sig.genes$logFC)),
    ]

chordSA <- chord_dat(data = circSA, genes = soCCE.res.sig.genes)

# Need to filter to something visually informative (based on order beta)
# Limit has to number, the first is the minimnum amount of terms need to be
# assigned to a gene and the second the minimum genes that must be assigned
# to a term. I choose 3 and 3 since this does not overcrowd the plot and
# cleanly highlights oxidative activity as well as iron ion binding.
goSAplot <- GOChord(chordSA, space = 0.02, gene.order = "logFC", gene.space = 0.25,
    gene.size = 3, process.label = 5, limit = c(3, 3), nlfc = 1)
goSAplot

## Warning: Using size for a discrete variable is not advised.

## Warning: Removed 13 rows containing missing values (geom_point).
```



Plot counts of GO terms in SE and CCE. Also show fold enrichment of terms. This is a little confusing, I basically remake a dataframe from the DAVID output that retains counts of genes mapping to each GO term. I didn't do this before since GOPlot doesn't like it.

```
# Make dataframes
soSE.res.sig.david.counts <- subset(soCCE.res.sig.f.david, Benjamini <= 0.05)
soCCE.res.sig.david.counts <- subset(soCCE.res.sig.f.david, Benjamini <= 0.05)

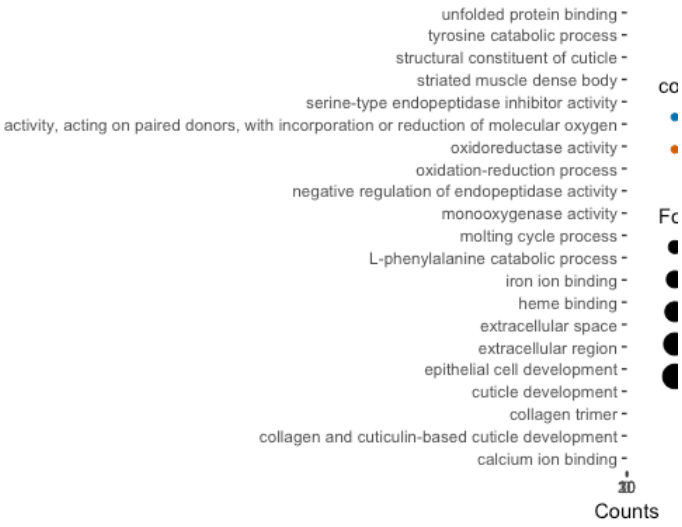
# Add conditions
soSE.res.sig.david.counts[, "condition"] <- rep("SE", length(rownames(soSE.res.sig.david.counts)))
soCCE.res.sig.david.counts[, "condition"] <- rep("CCE", length(rownames(soCCE.res.sig.d
```

```
avid.counts)))

# Combine dataframes
res.sig.david.combined <- rbind(soSE.res.sig.david.counts, soCCE.res.sig.david.counts)

# Remove GO term ID, can just comment this line out to retain ID
res.sig.david.combined[, "Term"] <- gsub(".*~", "", as.matrix(res.sig.david.combined[,
    "Term"]))

# Plot
ggplot(res.sig.david.combined, aes(Term, Count, fill = condition, colour = condition,
    size = Fold.Enrichment)) + geom_point() + theme(axis.text.x = element_text(angle =
-90,
    hjust = 0, vjust = 0.5)) + scale_colour_manual(values = c("#0072B2", "#D55E00")) +
    coord_flip() + scale_x_discrete(position = "right") + xlab("") + ylab("Counts") +
    ylim(0, 30) + theme_bw()
```




```
# Write out table of GO term for supplementary mats
res.sig.david.combined.supp.table <- res.sig.david.combined[, c("Term", "PValue",
    "Genes", "Fold.Enrichment", "Benjamini", "condition")]
write.csv(res.sig.david.combined.supp.table, "~/Documents/King_Lab/Masters_thesis/RNASe
q/sleuth/csv/res.sig.david.combined.supp.table.csv")
```

Same but for direct comparison between CCE and SE

```
ccese.res.david <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/cces
e.res.sig.func.annotation.csv")

ccese.res.sig.david <- GOplotdatDavid(ccese.res.david)
```

After running this I see that only one term remains significant. For plotting purposes I need to highlight at least two terms (circos won't plot otherwise), also it is interesting to see the close to marginally significant p = 0.07 that maps to defense factors, which is in fact an ancestor of innate immunity. Change our function slightly for this purpose and be clear about this in figure and results.

```
GOplotdatDavidCCESE <- function(soSE.res.sig.david) {
    Category <- soSE.res.sig.david$Category
    ID <- soSE.res.sig.david$Term
    ID <- gsub("~.*", "", ID)
    Term <- soSE.res.sig.david$Term
    Term <- gsub(".*~", "", Term)
    Genes <- soSE.res.sig.david$Genes
```

```
    adj_pval <- soSE.res.sig.david$Benjamini
    gopdat <- data.frame(Category, ID, Term, Genes, adj_pval)
    # CHANGE HERE
    gopdat <- subset(gopdat, adj_pval <= 0.08)
    return(gopdat)
}

ccese.res.sig.david <- GOplotdatDavidCCESE(ccese.res.david)
ccese.res.sig.david

##           Category         ID                   Term
## 1 GOTERM_BP_DIRECT GO:0045087 innate immune response
## 2 GOTERM_BP_DIRECT GO:0006952       defense response
##
                                    Genes
## 1 DOD-17, F56A4.2, LYS-1, B0024.4, K08D8.5, Y47H9C.1, DOD-22, CNC-6, F54B8.4, CLEC-6
7, CLEC-209, C17H12.8, CLEC-186, F54D5.4
## 2
   ILYS-3, FMO-2, VHP-1, B0024.4, DOD-22
##       adj_pval
## 1 0.0000000438
## 2 0.0727770150
```

Now continue to make circos plot

```
ccese.res.sig.genelist <- GOplotdatGeneList(ccese.res.sig)
ccese.process <- ccese.res.sig.david$Term
circccese <- circle_dat(ccese.res.sig.david, ccese.res.sig.genelist)
genes <- circccese$genes
logFC <- circccese$logFC
ccese.res.sig.genes <- data.frame(genes, logFC)
ccese.res.sig.genes <- unique(ccese.res.sig.genes)
ccese.res.sig.genes <- ccese.res.sig.genes[order(-abs(ccese.res.sig.genes$logFC)),
    ]

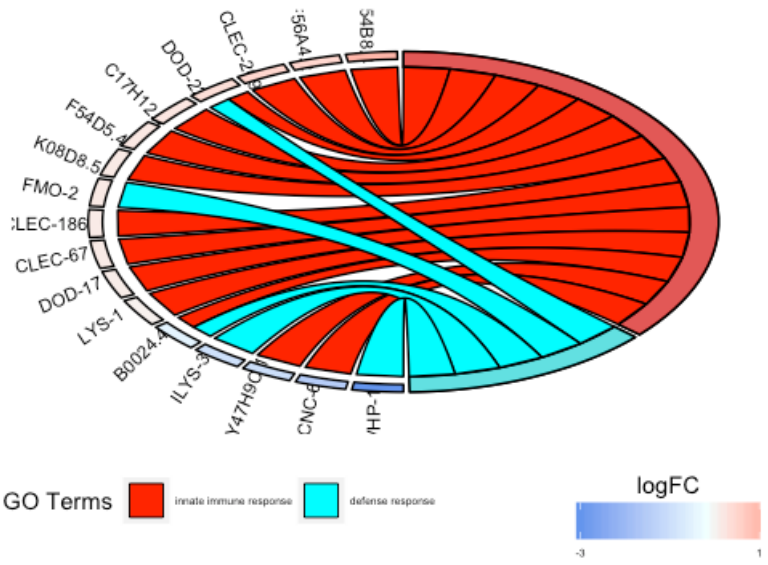
chordccese <- chord_dat(data = circccese, genes = ccese.res.sig.genes)


goCCESEplot <- GOChord(chordccese, space = 0.02, gene.order = "logFC", gene.space = 0.2
5,
    gene.size = 3, process.label = 5, nlfc = 1)

goCCESEplot

## Warning: Using size for a discrete variable is not advised.

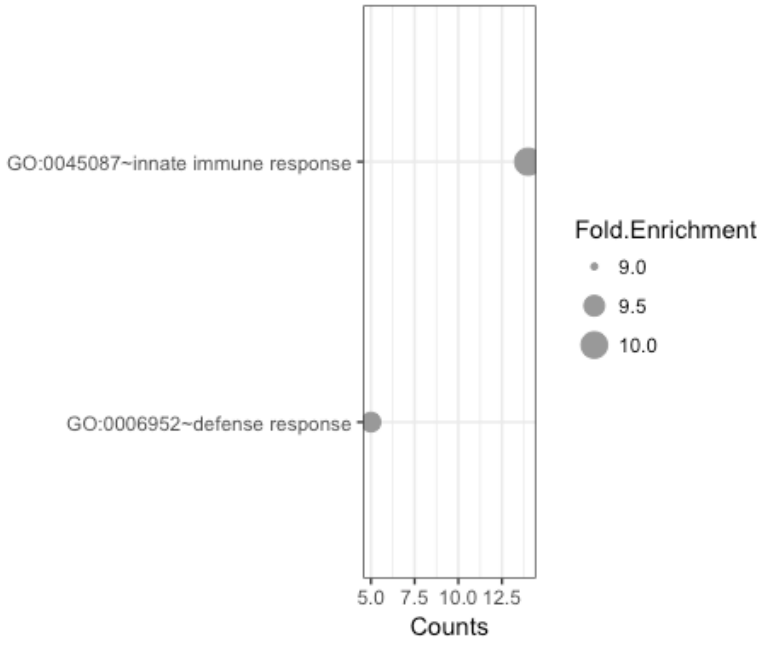
## Warning: Removed 2 rows containing missing values (geom_point).
```



Again, plot counts

```
ccese.res.david <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/ccese.res.sig.func.annotation.csv")
# Doing 0.08 so I can show defense factors
ccese.res.david.sig.counts <- subset(ccese.res.david, Benjamini <= 0.08)

ggplot(ccese.res.david.sig.counts, aes(Term, Count, size = Fold.Enrichment)) +
    geom_point(colour = "#999999") + theme(axis.text.x = element_text(angle = -90,
    hjust = 0, vjust = 0.5)) + coord_flip() + scale_x_discrete(position = "right") +
    xlab("") + ylab("Counts") + theme_bw() + scale_size(limits = c(9, 10.5),
    breaks = c(9, 9.5, 10))
```



Now I write out a table of the DEGs that map to these GO terms. Writing out the table with info from the full dataset and not just gene names so I have p-value info and WBGene ID

```
go_ccese <- as.data.frame(chordccese)
toupper(ccese.res.sig$ext_gene)

ccese.res.sig$ext_gene <- toupper(ccese.res.sig$ext_gene)
cce_go_degs <- ccese.res.sig[ccese.res.sig$ext_gene %in% rownames(go_ccese),
    ]
write.csv(cce_go_degs, "~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/ccese_go_
degs.csv")
```

Comparing AE to OP50

Comparing what proportion of DEGs were also observed in Wong et al. 2007 microarray study.

```
# read in csv with DEGs from wong et al. 2007
wong.efaec <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/ancestor_v_op
50/wong_efaec_diff_expressed.csv")
```

What percent of genes observed to differentially express with E. faecalis from Wong 2007 do I also observe?

```
inter <- intersect(wong.efaec$WB.Gene.ID, sopp.res.sig$ens_gene)
length(inter)/length(wong.efaec$WB.Gene.ID)

## [1] 0.6534181
```

What proportion of these changed in the same direction?

```
sopp.res.sig.up <- subset(sopp.res.sig, b > 0)
sopp.res.sig.down <- subset(sopp.res.sig, b < 0)
sopp.res.sig.inter <- sopp.res.sig[sopp.res.sig$ens_gene %in% inter, ]
sopp.res.sig.inter.up <- subset(sopp.res.sig.inter, b > 0)
sopp.res.sig.inter.down <- subset(sopp.res.sig.inter, b < 0)

wong.efaec.inter <- wong.efaec[wong.efaec$WB.Gene.ID %in% inter, ]
wong.efaec.inter.up <- subset(wong.efaec.inter, up_down == "up")
wong.efaec.inter.down <- subset(wong.efaec.inter, up_down == "down")
```

From Wong et al. (2007), highlight the genes they do. Most of these genes are pathogenesis related. From Figure 3. Common expression genes

```
wong.common.list <- c("asp-1", "asp-3", "asp-5", "asp-6", "clec-63", "clec-65",
    "clec-67", "acdh-1", "acdh-2", "ech-6", "pmt-2", "npp-13", "lys-1")

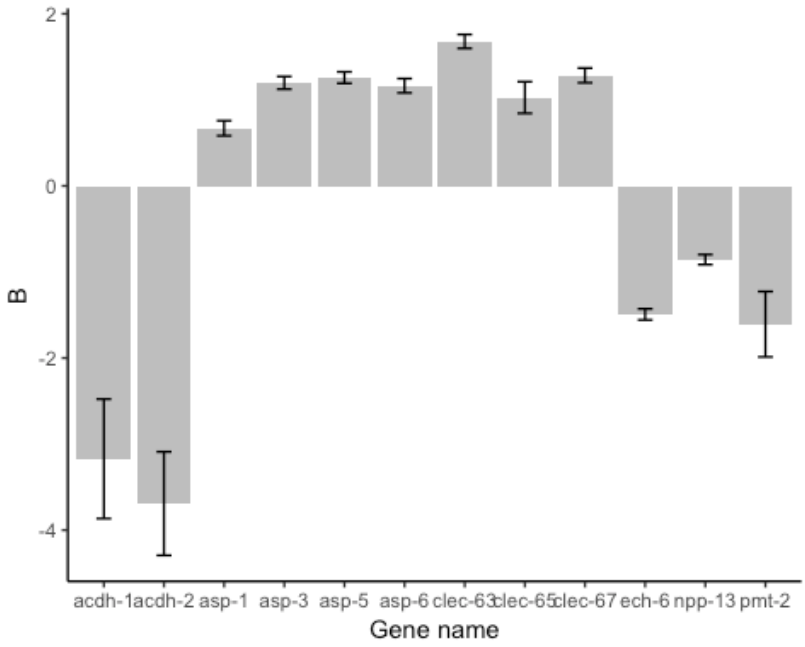
sopp.res.sig.wong.common <- sopp.res.sig[sopp.res.sig$ext_gene %in% wong.common.list,
    ]

# Multiple transcripts map to same gene so I take average effect
sopp.res.sig.wong.common.agg <- aggregate(sopp.res.sig.wong.common, list(sopp.res.sig.w
ong.common$ext_gene),
    FUN = mean, na.rm = FALSE)

limits <- aes(ymax = sopp.res.sig.wong.common.agg$b + sopp.res.sig.wong.common.agg$se_b
,
    ymin = sopp.res.sig.wong.common.agg$b - sopp.res.sig.wong.common.agg$se_b)

wong.comp <- ggplot(sopp.res.sig.wong.common.agg, aes(x = Group.1, y = b, fill = Group.
1)) +
    geom_bar(position = "dodge", fill = "grey", stat = "identity") + theme_classic() +
    geom_errorbar(limits, position = "dodge", width = 0.25) + xlab("Gene name") +
    ylab("B")
```

```
wong.comp
```



Plot counts of GO terms in AE. Also show fold enrichment of terms

```
write.csv(unique(na.omit(sopp.res.sig$ext_gene)), "~/Documents/King_Lab/Masters_thesis/
RNASeq/sleuth/ancestor_v_op50/efaecalis_res_sig_unique_extgene.csv",
    row.names = FALSE)

efaecalis_res_david <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/ance
stor_v_op50/efaecalis.res.sig.func.annotation.csv")
efaecalis_res_david.sig <- subset(efaecalis_res_david, Benjamini <= 0.05)
dim(efaecalis_res_david.sig)

## [1] 99 13

efaec_counts <- ggplot(efaecalis_res_david.sig, aes(Term, Count, size = Fold.Enrichment
)) +
    geom_point(colour = "#999999") + theme(axis.text.x = element_text(angle = -90,
    hjust = 0, vjust = 0.5)) + coord_flip() + scale_x_discrete(position = "right") +
    xlab("") + ylab("Counts") + theme_bw()
```

How many genes are related to functionally enriched GO terms?

```
length(unique(unlist(strsplit(as.character(efaecalis_res_david.sig$Genes), ","))))

## [1] 7503
```

How many functionally enriched GO terms are there?

```
length((efaecalis_res_david.sig$Term))

## [1] 99
```

Fold enrichment range, average and SE?

```
range(efaecalis_res_david.sig$Fold.Enrichment)

## [1] 1.071016 1.504365
```

```
mean(efaecalis_res_david.sig$Fold.Enrichment)

## [1] 1.267928

se <- function(x) sd(x)/sqrt(length(x))
se(efaecalis_res_david.sig$Fold.Enrichment)

## [1] 0.01133925
```

Order by counts. What three enriched functions had the most genes associated?

```
efaecalis_res_david.sig[order(-efaecalis_res_david.sig$Count), ][1:3, ]$Term
```

Specificity and generality of differentially expressed genes

First load datasets from different papers.

```
wong_all_genes <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/wong_all_spec_diff.csv")
wong_all_genes.up <- wong_all_genes[(wong_all_genes$up_down == "up"), ]
wong_all_genes.down <- wong_all_genes[(wong_all_genes$up_down == "down"), ]

troemel_pa <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/pa_deg_troemel_2006.csv")
troemel_pa.up <- troemel_pa[(troemel_pa$Fold.Change > 0), ]
troemel_pa.down <- troemel_pa[(troemel_pa$Fold.Change < 0), ]

iraz_saureus <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/saureus_irazoqui_et_al_2010.csv")
iraz_saureus.up <- iraz_saureus[(iraz_saureus$Fold.Change > 0), ]
iraz_saureus.down <- iraz_saureus[(iraz_saureus$Fold.Change < 0), ]
```

E. faecalis specific genes

For E. faecalis specific genes, I use the E. faecalis gene set thats generated from Wong et al. using microarrays. In theory I could use our E. faecalis set of differentially expressed genes but I used RNASeq tech and are comparing to papers that used microarrays. Such a comparison would be biased towards identifying more E. faecalis specific transcript since RNASeq allows us to quanitfy novel and rare transcripts and detect transcripts across a broader range than microarrays.

Pulling DEG list from Wong et al. investigating four different colonizers (S. marcescens; E. faecalis; Erwinia carotovora; Photorhabdus luminescens); Irazoqiu et al. (S. aureus); and Troemel et al. (PA14). I then pull those that are unique to E. faecalis (using microarrays), and see which are found as significant in our treatments.

Need to do up/down seperate because if a gene is up regulated with E. faecalis exposure and down regualted with another bug, for instance S. marcesens, then it is still E. faecalis specific since the response is specific.

```
# subset to genes found in non ef species non ef up
wong_all_genes.non_ef.up <- wong_all_genes.up[!(wong_all_genes.up$species ==
    "ef"), ]
# non ef down
wong_all_genes.non_ef.down <- wong_all_genes.down[!(wong_all_genes.up$species ==
    "ef"), ]
# subset to genes found in ef
wong.ef <- wong_all_genes[(wong_all_genes$species == "ef"), ]
# ef up
wong.ef.up <- wong.ef[(wong.ef$up_down == "up"), ]
# ef down
wong.ef.down <- wong.ef[(wong.ef$up_down == "down"), ]
```

```
# Subset to ef specific genes in up/down
wong.ef.spec.genes.up <- subset(wong.ef.up, !wong.ef.up$wormbase %in% wong_all_genes.no
n_ef.up$wormbase)$wormbase
wong.ef.spec.genes.down <- subset(wong.ef.down, !wong.ef.down$wormbase %in%
    wong_all_genes.non_ef.down$wormbase)$wormbase


# E. faecalis specific genes in SE compared to ancestor
soSE.res.sig.up <- soSE.res.sig[(soSE.res.sig$b > 0), ]
soSE.res.sig.down <- soSE.res.sig[(soSE.res.sig$b < 0), ]
se_ef_spec_up <- subset(soSE.res.sig.up, soSE.res.sig.up$ens_gene %in% wong.ef.spec.gen
es.up) %>%
    subset(., !.$target_id %in% troemel_pa.up$Gene.ID) %>% subset(., !.$target_id %in%
    iraz_saureus.up$Cosmid.Name) %>% unique() %>% na.omit()

se_ef_spec_down <- subset(soSE.res.sig.down, soSE.res.sig.down$ens_gene %in%
    wong.ef.spec.genes.down) %>% subset(., !.$target_id %in% troemel_pa.down$Gene.ID) %
>%
    subset(., !.$target_id %in% iraz_saureus.down$Cosmid.Name) %>% unique() %>%
    na.omit()

se_ef_spec <- rbind(se_ef_spec_up, se_ef_spec_down)

# E. faecalis specific genes in CCE compared to ancestor
soCCE.res.sig.up <- soCCE.res.sig[(soCCE.res.sig$b > 0), ]
soCCE.res.sig.down <- soCCE.res.sig[(soCCE.res.sig$b < 0), ]

cce_ef_spec_up <- subset(soCCE.res.sig.up, soCCE.res.sig.up$ens_gene %in% wong.ef.spec.
genes.up) %>%
    subset(., !.$target_id %in% troemel_pa.up$Gene.ID) %>% subset(., !.$target_id %in%
    iraz_saureus.up$Cosmid.Name) %>% unique() %>% na.omit()

cce_ef_spec_down <- subset(soCCE.res.sig.down, soCCE.res.sig.down$ens_gene %in%
    wong.ef.spec.genes.down) %>% subset(., !.$target_id %in% troemel_pa.down$Gene.ID) %
>%
    subset(., !.$target_id %in% iraz_saureus.down$Cosmid.Name) %>% unique() %>%
    na.omit()

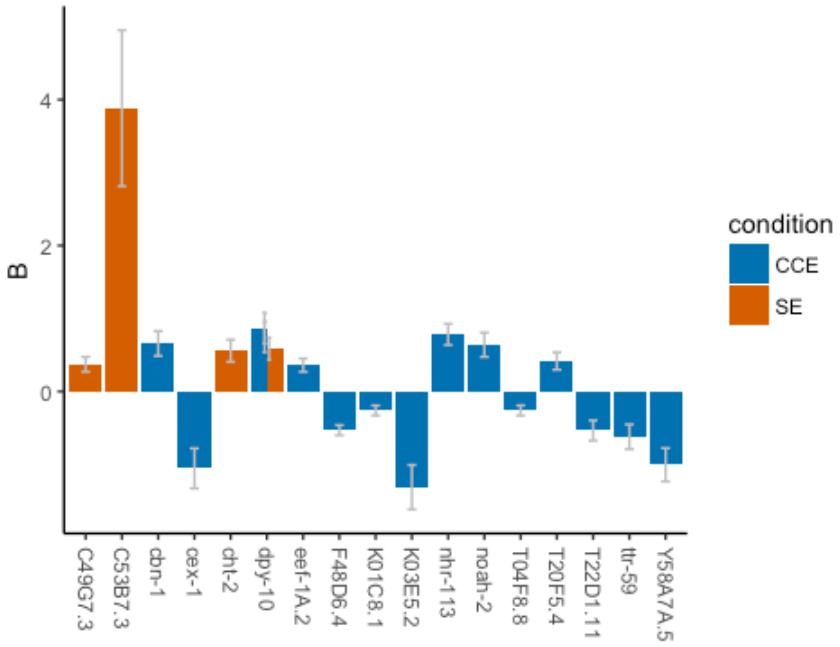
cce_ef_spec <- rbind(cce_ef_spec_up, cce_ef_spec_down)

se_cce_ef_spec_comb <- rbind(cce_ef_spec, se_ef_spec)


se_cce_ef_spec_comb_limits <- aes(ymax = se_cce_ef_spec_comb$b + se_cce_ef_spec_comb$se
_b,
    ymin = se_cce_ef_spec_comb$b - se_cce_ef_spec_comb$se_b)


se_cce_ef_spec_comb_p <- ggplot(se_cce_ef_spec_comb, aes(x = ext_gene, y = b,
    fill = condition)) + geom_bar(position = "dodge", stat = "identity") + theme_classi
c() +
    xlab(" ") + ylab("B") + geom_errorbar(se_cce_ef_spec_comb_limits, position = "dodge
",
    width = 0.25, color = "grey") + scale_fill_manual(values = c("#0072B2",
    "#D55E00")) + theme(axis.text.x = element_text(angle = -90, hjust = 0, vjust = 0.5)
)

se_cce_ef_spec_comb_p
```



Find genes that are differentialy expressed by S. aureus exposure and overlap with E. faecalis induced genes.

Use intersect instead? Why public name and not target ID?

```
# S. aureus & E. faecalis common genes
iraz_wong_ef_overlap_up <- subset(iraz_saureus.up, iraz_saureus.up$Cosmid.Name %in%
    wong.ef.up$ext_gene)
iraz_wong_ef_overlap_down <- subset(iraz_saureus.down, iraz_saureus.down$Cosmid.Name %i
n%
    wong.ef.down$ext_gene)

# Subset common genes similarly differentially expressed in CCE exposure,
# here I use public name because with target_id I miss a transcript that
# has multiple types (e.g., .1 , .2)
sa_ef_cmmn_cce <- rbind(subset(soCCE.res.sig.up, soCCE.res.sig.up$ext_gene %in%
    iraz_wong_ef_overlap_up$Public.Name), subset(soCCE.res.sig.down, soCCE.res.sig.down
$ext_gene %in%
    iraz_wong_ef_overlap_down$Public.Name))

# Subset common genes similarly differentially expressed in SE exposure
sa_ef_cmmn_se <- rbind(subset(soSE.res.sig.up, soSE.res.sig.up$ext_gene %in%
    iraz_wong_ef_overlap_up$Public.Name), subset(soSE.res.sig.down, soSE.res.sig.down$e
xt_gene %in%
    iraz_wong_ef_overlap_down$Public.Name))

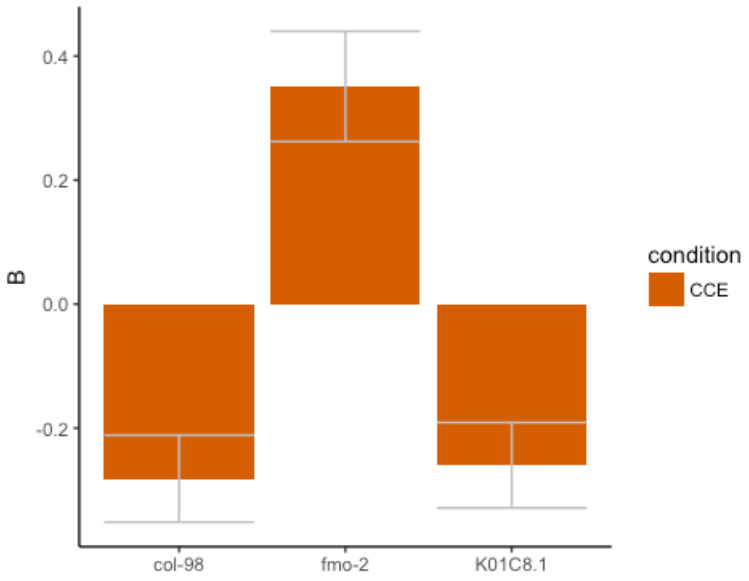

sa_ef_cmmn <- rbind(sa_ef_cmmn_cce, sa_ef_cmmn_se)

sa_ef_cmmn_limits <- aes(ymax = sa_ef_cmmn$b + sa_ef_cmmn$se_b, ymin = sa_ef_cmmn$b -
    sa_ef_cmmn$se_b)


sa_ef_cmmn_p <- ggplot(sa_ef_cmmn, aes(x = ext_gene, y = b, fill = condition)) +
    geom_bar(position = "dodge", stat = "identity") + theme_classic() + xlab(" ") +
    ylab("B") + geom_errorbar(sa_ef_cmmn_limits, position = "dodge", widranth = 0.25,
    color = "grey") + scale_fill_manual(values = c("#D55E00"))
```

```
## Warning: Ignoring unknown parameters: widranth
```

```
sa_ef_cmmn_p
```



S. aureus specific genes. Here, to be extra conservative, I can remove those that I identified in our RNASeq DEGs with E. faecalis.

```
sopp.res.sig.up <- sopp.res.sig[(sopp.res.sig$b > 0), ]
sopp.res.sig.down <- sopp.res.sig[(sopp.res.sig$b < 0), ]

iraz_saureus.up.spcfc <- subset(iraz_saureus.up, !iraz_saureus.up$Cosmid.Name %in%
    wong_all_genes.up$ext_gene) %>% subset(., !.$Cosmid.Name %in% troemel_pa.up$Gene.ID
) %>%
    subset(., !.$Cosmid.Name %in% sopp.res.sig.up$target_id) %>% unique()


iraz_saureus.down.spcfc <- subset(iraz_saureus.down, !iraz_saureus.down$Cosmid.Name %in
%
    wong_all_genes.down$ext_gene) %>% subset(., !.$Cosmid.Name %in% troemel_pa.down$Gen
e.ID) %>%
    subset(., !.$Cosmid.Name %in% sopp.res.sig.down$target_id) %>% unique()

sa_spcfc_se <- rbind(subset(soSE.res.sig.up, soSE.res.sig.up$ext_gene %in% iraz_saureus
.up.spcfc$Public.Name),
    subset(soSE.res.sig.down, soSE.res.sig.down$ext_gene %in% iraz_saureus.down.spcfc$P
ublic.Name))

sa_spcfc_cce <- rbind(subset(soCCE.res.sig.up, soCCE.res.sig.up$ext_gene %in%
    iraz_saureus.up.spcfc$Public.Name), subset(soCCE.res.sig.down, soCCE.res.sig.down$e
xt_gene %in%
    iraz_saureus.down.spcfc$Public.Name))

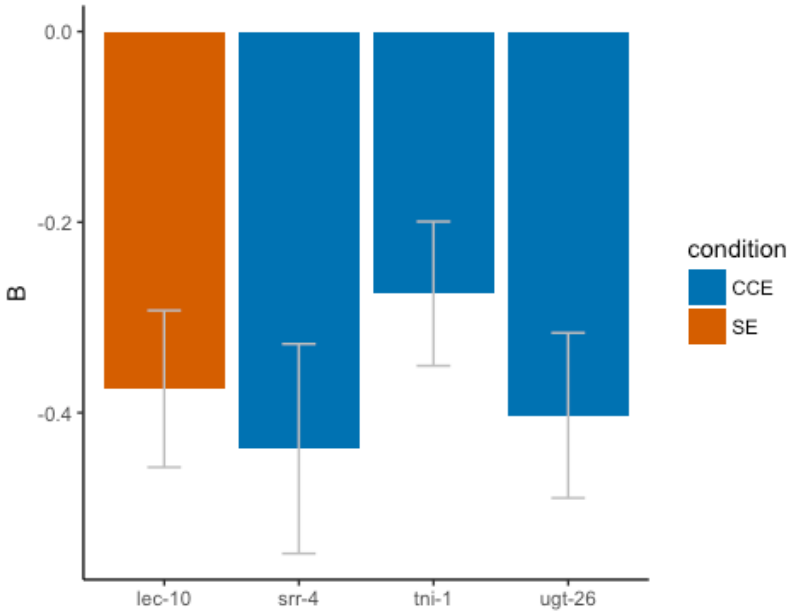
sa_spcfc <- rbind(sa_spcfc_se, sa_spcfc_cce)

sa_limits <- aes(ymax = sa_spcfc$b + sa_spcfc$se_b, ymin = sa_spcfc$b - sa_spcfc$se_b)


sa_spcfc_p <- ggplot(sa_spcfc, aes(x = ext_gene, y = b, fill = condition)) +
    geom_bar(position = "dodge", stat = "identity") + theme_classic() + xlab(" ") +
    ylab("B") + geom_errorbar(sa_limits, position = "dodge", width = 0.25, color = "gre
```

```
y") +
    scale_fill_manual(values = c("#0072B2", "#D55E00"))

sa_spcfc_p
```



Make list of general pathogenesis genes: Query Kim et al. for generalized mechanisms, particularly those E. faecalis and S. aureus general.

Wong et al. revealed strong shared response of 22 genes, defineing them as generalized responses. First let's confirm and see how many of these E. faecalis regulates over OP50, then see how many evolved response with SE and CCE. The following are taken from Table 1 in Wong et al. 2007.

```
wong_gen <- read.csv("~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/csv/wong_general_pathogenesis.csv")

wong_gen_up <- wong_gen[(wong_gen$X == "up"), ]
wong_gen_down <- wong_gen[(wong_gen$X == "down"), ]

se_gen <- rbind(subset(soSE.res.sig.up, soSE.res.sig.up$target_id %in% wong_gen_up$Sequence.name),
    subset(soSE.res.sig.down, soSE.res.sig.down$target_id %in% wong_gen_down$Sequence.name))

cce_gen <- rbind(subset(soCCE.res.sig.up, soCCE.res.sig.up$ext_gene %in% wong_gen_up$Gene.name),
    subset(soCCE.res.sig.down, soCCE.res.sig.down$target_id %in% wong_gen_down$Sequence.name))
```

No genes defined as general by Wong et al. definition. Alternatively, I can find the intersection of all diff expressed genes from diff papers and use that as different definition

```
devtools::session_info()
## Session info -------------------------------------------------------------
##  setting  value
##  version  R version 3.4.0 (2017-04-21)
##  system   x86_64, darwin15.6.0
```

```
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/Los_Angeles
##  date     2017-06-01
## Packages -----------------------------------------------------------------
##  package        * version  date       source
##  AnnotationDbi    1.38.0   2017-04-25 Bioconductor
##  assertthat       0.2.0    2017-04-11 cran (@0.2.0)
##  backports        1.0.5    2017-01-18 CRAN (R 3.4.0)
##  base           * 3.4.0    2017-04-21 local
##  Biobase          2.36.2   2017-05-04 Bioconductor
##  BiocGenerics     0.22.0   2017-04-25 Bioconductor
##  biomaRt        * 2.32.0   2017-04-26 Bioconductor
##  bitops           1.0-6    2013-08-17 CRAN (R 3.4.0)
##  codetools        0.2-15   2016-10-05 CRAN (R 3.4.0)
##  colorspace       1.3-2    2016-12-14 CRAN (R 3.4.0)
##  compiler         3.4.0    2017-04-21 local
##  data.table       1.10.4   2017-02-01 CRAN (R 3.4.0)
##  datasets       * 3.4.0    2017-04-21 local
##  DBI              0.6-1    2017-04-01 CRAN (R 3.4.0)
##  devtools       * 1.13.1   2017-05-13 CRAN (R 3.4.0)
##  digest           0.6.12   2017-01-27 CRAN (R 3.4.0)
##  dplyr          * 0.5.0    2016-06-24 cran (@0.5.0)
##  evaluate         0.10     2016-10-11 CRAN (R 3.4.0)
##  formatR          1.5      2017-04-25 CRAN (R 3.4.0)
##  futile.logger  * 1.4.3    2016-07-10 CRAN (R 3.4.0)
##  futile.options   1.0.0    2010-04-06 CRAN (R 3.4.0)
##  ggdendro       * 0.1-20   2016-04-27 CRAN (R 3.4.0)
##  ggplot2        * 2.2.1    2016-12-30 CRAN (R 3.4.0)
##  GOplot         * 1.0.2    2016-03-30 CRAN (R 3.4.0)
##  graphics       * 3.4.0    2017-04-21 local
##  grDevices      * 3.4.0    2017-04-21 local
##  grid           * 3.4.0    2017-04-21 local
##  gridExtra      * 2.2.1    2016-02-29 CRAN (R 3.4.0)
##  gtable           0.2.0    2016-02-26 CRAN (R 3.4.0)
##  htmltools        0.3.6    2017-04-28 CRAN (R 3.4.0)
##  httpuv           1.3.3    2015-08-04 cran (@1.3.3)
##  IRanges          2.10.1   2017-05-11 Bioconductor
##  knitr            1.15.1   2016-11-22 CRAN (R 3.4.0)
##  lambda.r         1.1.9    2016-07-10 CRAN (R 3.4.0)
##  lazyeval         0.2.0    2016-06-12 CRAN (R 3.4.0)
##  magrittr         1.5      2014-11-22 CRAN (R 3.4.0)
##  MASS             7.3-47   2017-02-26 CRAN (R 3.4.0)
##  memoise          1.1.0    2017-04-21 CRAN (R 3.4.0)
##  methods        * 3.4.0    2017-04-21 local
##  mime             0.5      2016-07-07 CRAN (R 3.4.0)
##  munsell          0.4.3    2016-02-13 CRAN (R 3.4.0)
##  parallel         3.4.0    2017-04-21 local
##  plyr           * 1.8.4    2016-06-08 CRAN (R 3.4.0)
##  R6               2.2.1    2017-05-10 CRAN (R 3.4.0)
##  RColorBrewer   * 1.1-2    2014-12-07 CRAN (R 3.4.0)
##  Rcpp             0.12.10  2017-03-19 CRAN (R 3.4.0)
##  RCurl            1.95-4.8 2016-03-01 CRAN (R 3.4.0)
##  reshape2         1.4.2    2016-10-22 CRAN (R 3.4.0)
##  rhdf5            2.20.0   2017-04-25 Bioconductor
##  rmarkdown        1.5      2017-04-26 CRAN (R 3.4.0)
##  rprojroot        1.2      2017-01-16 CRAN (R 3.4.0)
##  RSQLite          1.1-2    2017-01-08 CRAN (R 3.4.0)
##  S4Vectors        0.14.1   2017-05-11 Bioconductor
##  scales           0.4.1    2016-11-09 CRAN (R 3.4.0)
```

```
##  shiny          * 1.0.3    2017-04-26 cran (@1.0.3)
##  sleuth         * 0.28.1   2017-05-09 Github (pachterlab/sleuth@048f055)
##  stats          * 3.4.0    2017-04-21 local
##  stats4           3.4.0    2017-04-21 local
##  stringi          1.1.5    2017-04-07 CRAN (R 3.4.0)
##  stringr          1.2.0    2017-02-18 CRAN (R 3.4.0)
##  tibble           1.3.0    2017-04-01 CRAN (R 3.4.0)
##  tidyr            0.6.2    2017-05-04 cran (@0.6.2)
##  tools            3.4.0    2017-04-21 local
##  utils          * 3.4.0    2017-04-21 local
##  VennDiagram    * 1.6.17   2016-04-18 CRAN (R 3.4.0)
##  withr            1.0.2    2016-06-20 CRAN (R 3.4.0)
##  XML              3.98-1.7 2017-05-03 CRAN (R 3.4.0)
##  xtable           1.8-2    2016-02-05 CRAN (R 3.4.0)
##  yaml             2.1.14   2016-11-12 CRAN (R 3.4.0)
##  zlibbioc         1.22.0   2017-04-25 Bioconductor
```

Supplementary file 4. R Markdown file outlining 16S rRNA read processing and analyses.

Load libraries and add functions

Functions added are taken from github, either this phyloseq thread or kdauria github code.

```
library(dada2); packageVersion("dada2")
```

```
## Loading required package: Rcpp
```

```
## [1] '1.4.0'
```

```
library(phyloseq); packageVersion("phyloseq")
```

```
## [1] '1.20.0'
```

```
library(ggplot2); packageVersion("ggplot2")
```

```
## [1] '2.2.1'
```

```
library(ape); packageVersion("ape")
```

```
## [1] '4.1'
```

```
library(plotly); packageVersion("plotly")
```

```
library(vegan); packageVersion("vegan")

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.4-3

## [1] '2.4.3'

library(limma); packageVersion("limma")

## [1] '3.32.2'

library(data.table); packageVersion("data.table")

## [1] '1.10.4'

library(plyr); packageVersion("plyr")

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:plotly':
##
##     arrange, mutate, rename, summarise

## [1] '1.8.4'

se <- function(x) sd(x)/sqrt(length(x))

fast_melt = function(physeq){
  # supports "naked" otu_table as `physeq` input.
  otutab = as(otu_table(physeq), "matrix")
  if(!taxa_are_rows(physeq)){otutab <- t(otutab)}
  otudt = data.table(otutab, keep.rownames = TRUE)
  setnames(otudt, "rn", "taxaID")
  # Enforce character taxaID key
  otudt[, taxaIDchar := as.character(taxaID)]
  otudt[, taxaID := NULL]
  setnames(otudt, "taxaIDchar", "taxaID")
  # Melt count table
  mdt = melt.data.table(otudt,
                        id.vars = "taxaID",
                        variable.name = "SampleID",
                        value.name = "count")
  # Remove zeroes, NAs
  mdt <- mdt[count > 0][!is.na(count)]
  # Calculate relative abundance
  mdt[, RelativeAbundance := count / sum(count), by = SampleID]
  if(!is.null(tax_table(physeq, errorIfNULL = FALSE))){
    # If there is a tax_table, join with it. Otherwise, skip this join.
    taxdt = data.table(as(tax_table(physeq, errorIfNULL = TRUE), "matrix"), keep.rownam
es = TRUE)
    setnames(taxdt, "rn", "taxaID")
    # Enforce character taxaID key
    taxdt[, taxaIDchar := as.character(taxaID)]
    taxdt[, taxaID := NULL]
    setnames(taxdt, "taxaIDchar", "taxaID")
    # Join with tax table
    setkey(taxdt, "taxaID")
    setkey(mdt, "taxaID")
    mdt <- taxdt[mdt]
```

```
  }
  return(mdt)
}

summarize_taxa = function(physeq, Rank, GroupBy = NULL){
  Rank <- Rank[1]
  if(!Rank %in% rank_names(physeq)){
    message("The argument to `Rank` was:\n", Rank,
            "\nBut it was not found among taxonomic ranks:\n",
            paste0(rank_names(physeq), collapse = ", "), "\n",
            "Please check the list shown above and try again.")
  }
  if(!is.null(GroupBy)){
    GroupBy <- GroupBy[1]
    if(!GroupBy %in% sample_variables(physeq)){
      message("The argument to `GroupBy` was:\n", GroupBy,
              "\nBut it was not found among sample variables:\n",
              paste0(sample_variables(physeq), collapse = ", "), "\n",
              "Please check the list shown above and try again.")
    }
  }
  # Start with fast melt
  mdt = fast_melt(physeq)
  if(!is.null(GroupBy)){
    # Add the variable indicated in `GroupBy`, if provided.
    sdt = data.table(SampleID = sample_names(physeq),
                     var1 = get_variable(physeq, GroupBy))
    setnames(sdt, "var1", GroupBy)
    # Join
    setkey(sdt, SampleID)
    setkey(mdt, SampleID)
    mdt <- sdt[mdt]
  }
  # Summarize
Nsamples = nsamples(physeq)
  summarydt = mdt[, list(meanRA = sum(RelativeAbundance)/Nsamples,
                         sdRA = sd(RelativeAbundance),
                         seRA = se(RelativeAbundance),
                         minRA = min(RelativeAbundance),
                         maxRA = max(RelativeAbundance)),
                  by = c(Rank, GroupBy)]
  return(summarydt)
}


# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)
```

```
  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

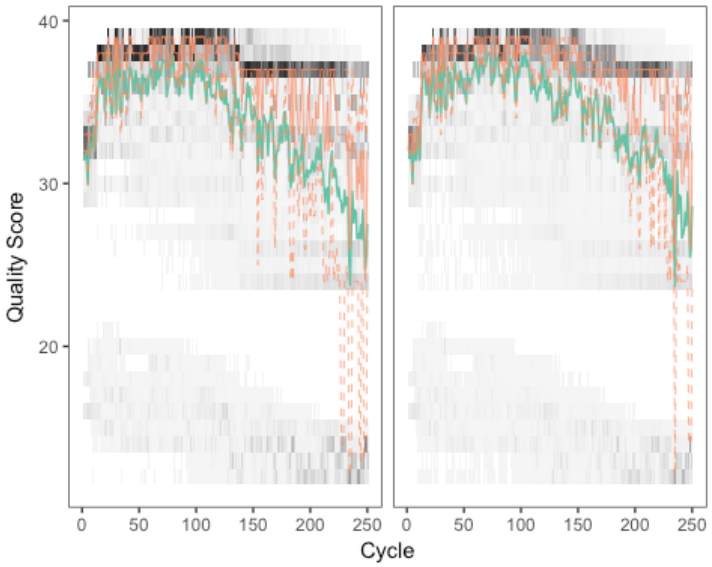
      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
set.seed(100)

path <- "~/Documents/King_Lab/Masters_thesis/16SrRNA/data"

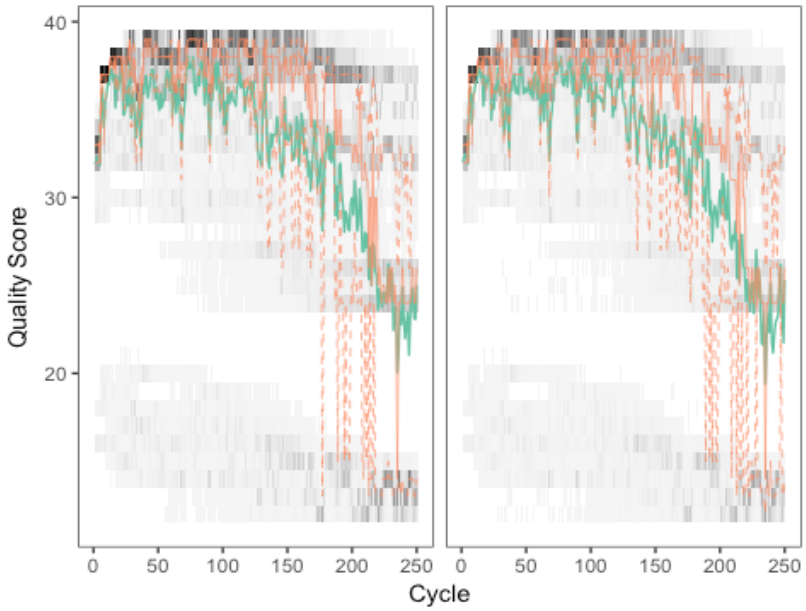
# Sort ensures forward/reverse reads are in same order
fnFs <- sort(list.files(path, pattern = "_R1_001.fastq"))
fnRs <- sort(list.files(path, pattern = "_R2_001.fastq"))
# Extract sample names, assuming filenames have format: SAMPLENAME_XXX.fastq
sample.names <- sapply(strsplit(fnFs, "\\."), function(x) x[1])

# Specify the full path to the fnFs and fnRs
fnFs <- file.path(path, fnFs)
fnRs <- file.path(path, fnRs)

plotQualityProfile(fnFs[1:2])
```




```
plotQualityProfile(fnRs[1:2])
```



#Filtering and trimming

```
filt_path <- file.path(path, "filtered") # Place filtered files in filtered/ subdirecto
ry
filtFs <- file.path(filt_path, paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(filt_path, paste0(sample.names, "_R_filt.fastq.gz"))

out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, truncLen=c(240,160),
              maxN=0, maxEE=c(2,2), truncQ=2, rm.phix=TRUE,
              compress=TRUE, multithread=TRUE)
```

Learn error rates

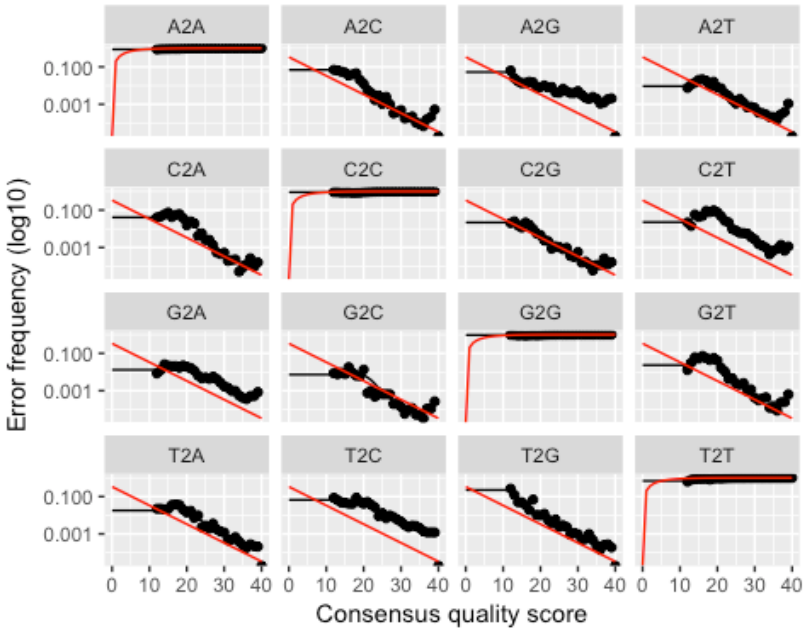
Learn error rates of forward reads

```
errF <- learnErrors(filtFs, multithread=TRUE)
```

Learn those of reverse reads

```
errR <- learnErrors(filtRs, multithread=TRUE)

plotErrors(errF, nominalQ=TRUE)
```



Dereplication

```
derepFs <- derepFastq(filtFs, verbose=TRUE)

derepRs <- derepFastq(filtRs, verbose=TRUE)

# Name the derep-class objects by the sample names
names(derepFs) <- sample.names
names(derepRs) <- sample.names
```

Sample inference

```
dadaFs <- dada(derepFs, err=errF, multithread=TRUE)

dadaRs <- dada(derepRs, err=errR, multithread=TRUE)
```

Merge paired end reads

```
mergers <- mergePairs(dadaFs, derepFs, dadaRs, derepRs, verbose=TRUE)
```

Construct sequence table

```
seqtab <- makeSequenceTable(mergers)

table(nchar(getSequences(seqtab)))

72    1    2    4   40 2076   75    4    1    3    1    1
```

Remove chimeras

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbose=TRUE)

## Identified 502 bimeras out of 2280 input sequences.

dim(seqtab.nochim)

## [1]   82 1778

sum(seqtab.nochim)/sum(seqtab)

## [1] 0.9539578

save(seqtab.nochim, file = "~/Documents/King_Lab/Masters_thesis/16SrRNA/seqtab.nochim.Rdata")
```

Track amount of reads maintaned throughout each step of pipeline

```
getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(mergers, getN), rowSums(seqtab),
    rowSums(seqtab.nochim))
colnames(track) <- c("input", "filtered", "denoised", "merged", "tabled", "nonchim")
rownames(track) <- sample.names
head(track)
```

Assign taxonomy

Assign taxonomy with DADA2's native implementation of RDP's naive Bayesian classifier, using the GreenGenes 13.8 release clustered at 97% identity, the same fasta previously used to assign *C. elegans* microbiota taxonomy.

```
taxa <- assignTaxonomy(seqtab.nochim, "~/Documents/King_Lab/Masters_thesis/16SrRNA/training_set/gg_13_8_train_set_97.fa.gz", multithread=TRUE)
unname(head(taxa))

save(taxa, file = "~/Documents/King_Lab/Masters_thesis/16SrRNA/taxa.Rdata")
```

Export unique sequences in fasta format for multiple alignment. Multiple alignment is conducted in terminal, using QIIME.

```
uniquesToFasta(getUniques(seqtab.nochim), "~/Documents/King_Lab/Masters_thesis/16SrRNA/unique_sequences.fasta")

MacQIIME Macintosh-24:16SrRNA $ perl -ane 'if(/\>/){$a++;print ">sq$a\n"}else{print;}' unique_sequences.fasta > unique_sequences_renamed.fasta
```

Use PyNAST to build multiple aligment via QIIME.

```
MacQIIME Macintosh-24:16SrRNA $ align_seqs.py -i unique_sequences_renamed.fasta -o align_seqs -p 60
```

Build phylogenetic tree

```
MacQIIME Macintosh-24:16SrRNA $ make_phylogeny.py -i align_seqs/unique_sequences_renamed_aligned.fasta -o sequence.tre
```

Phyloseq

```
samples.out <- rownames(seqtab.nochim)
write.csv(samples.out,"~/Documents/King_Lab/Masters_thesis/16SrRNA/samplesout.csv")

#read sample data csv
```

```
samdf <- read.csv("~/Documents/King_Lab/Masters_thesis/16SrRNA/sampledata.csv")
rownames(samdf) <- samdf$sample.out
samdf$plate <- as.factor(samdf$plate)
#samdf <- rbind(samdf[order(samdf$treatment),][1:30,],samdf[order(samdf$treatment),][36
:80,])
#samdf <- rbind(samdf[order(samdf$batch),][1:25,],samdf[order(samdf$batch),][51:75,])


#read in tree
phytpy <- read_tree("~/Documents/King_Lab/Masters_thesis/16SrRNA/sequence.tre")

#Make sequence names match tip labels
sqrep <- rep(1:dim(seqtab.nochim)[2])
sqrep <- paste("sq",sqrep,sep="")
colnames(seqtab.nochim) <- sqrep
rownames(taxa) <- sqrep
```

Remove NTCs and extraction control

```
#Make seperate ps objects for each plate
samdfp5 <- subset(samdf, plate == "5")
samdfp6 <- subset(samdf, plate == "6")
samdfp7 <- subset(samdf, plate == "7")

psp5 <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows=FALSE),
                 sample_data(samdfp5),
                 tax_table(taxa))
psp5 = prune_taxa(taxa_sums(psp5)>0, psp5)
ntcp5 <- subset_samples(psp5, treatment == "NTC")
ntcp5 <- prune_taxa(taxa_sums(ntcp5)>0,ntcp5)
alltaxp5 <- names(sort(taxa_sums(psp5),TRUE))
ntctaxp5 <- names(sort(taxa_sums(ntcp5),TRUE))
nontctaxp5 = alltaxp5[!(alltaxp5 %in% ntctaxp5)]
psp5nontx <- prune_taxa(nontctaxp5,psp5)


psp6 <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows=FALSE),
                 sample_data(samdfp6),
                 tax_table(taxa))
psp6 = prune_taxa(taxa_sums(psp6)>0, psp6)
psp6 = prune_taxa(taxa_sums(psp6)>0, psp6)
ntcp6 <- subset_samples(psp6, treatment == "NTC")
ntcp6 <- prune_taxa(taxa_sums(ntcp6)>0,ntcp6)
alltaxp6 <- names(sort(taxa_sums(psp6),TRUE))
ntctaxp6 <- names(sort(taxa_sums(ntcp6),TRUE))
nontctaxp6 = alltaxp6[!(alltaxp6 %in% ntctaxp6)]
psp6nontx <- prune_taxa(nontctaxp6,psp6)


psp7 <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows=FALSE),
                 sample_data(samdfp7),
                 tax_table(taxa))
psp7 = prune_taxa(taxa_sums(psp7)>0, psp7)
ntcp7 <- subset_samples(psp7, treatment == "NTC")
ntcp7 <- prune_taxa(taxa_sums(ntcp7)>0,ntcp7)
alltaxp7 <- names(sort(taxa_sums(psp7),TRUE))
ntctaxp7 <- names(sort(taxa_sums(ntcp7),TRUE))
nontctaxp7 = alltaxp7[!(alltaxp7 %in% ntctaxp7)]
psp7nontx <- prune_taxa(nontctaxp7,psp7)


ps <- merge_phyloseq(psp5nontx,psp6nontx,psp7nontx)
```

```
#Remove extraction control
ext.cont <- subset_samples(ps, sample.id == "ext.cont")
ext.cont <- prune_taxa(taxa_sums(ext.cont)>0,ext.cont)
alltaxa <- names(sort(taxa_sums(ps),TRUE))
alltaxa.ext.cont <- names(sort(taxa_sums(ext.cont),TRUE))
nonextcont.ps = alltaxa[!(alltaxa %in% alltaxa.ext.cont)]
ps <- prune_taxa(nonextcont.ps,ps)

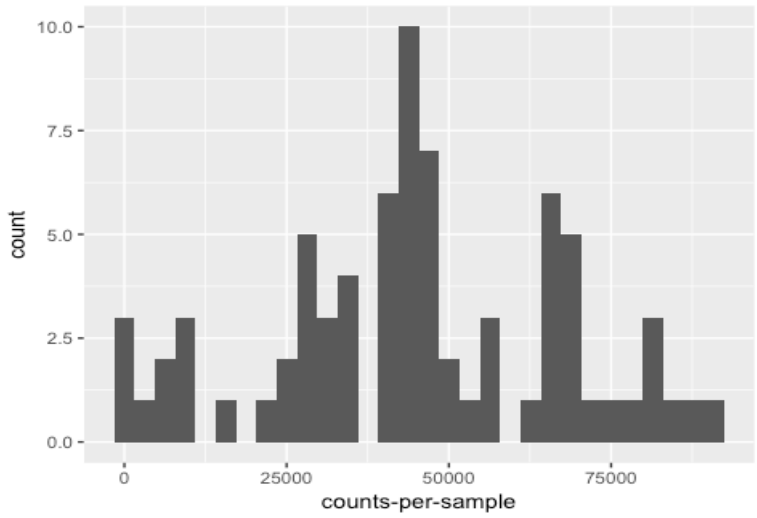

ps <- merge_phyloseq(ps,phytpy)

ps <- subset_samples(ps, treatment != "NTC")
ps <- subset_samples(ps, treatment != "ext_cont")
pssoil <- subset_samples(ps, treatment == "soil")
pssoil <- prune_taxa(taxa_sums(pssoil)>0,pssoil)
pssoil_tax <- data.frame(tax_table(pssoil))
ps <- subset_samples(ps, treatment != "soil")

save(ps, file = "~/Documents/King_Lab/Masters_thesis/16SrRNA/ps.Rdata")
#load(file = "~/Documents/King_Lab/Masters_thesis/16SrRNA/ps.Rdata")
```

```
Preprocessing and prefiltering
#Only retain bacteria and remove mitochondria and chloroplast
ps <- ps %>%
  subset_taxa(
    Kingdom == "k__Bacteria" &
    Family  != "f__mitochondria" &
    Class   != "c__Chloroplast"
  )

#Check distribution of counts in samples
qplot(rowSums(otu_table(ps))) +
  xlab("counts-per-sample")
```




```
#Handful of samples with < 15,000 reads while the average is ~50,000, remove these
sampsums <- as.data.frame(sample_sums(ps))
min(sample_sums(ps))

## [1] 330

mean(sample_sums(ps))

## [1] 44903.88
```

```r
ps <- prune_samples(sample_sums(ps) >= 15000, ps)

#How many RSVs on average in each sample?
mean(estimate_richness(ps)$Observed)

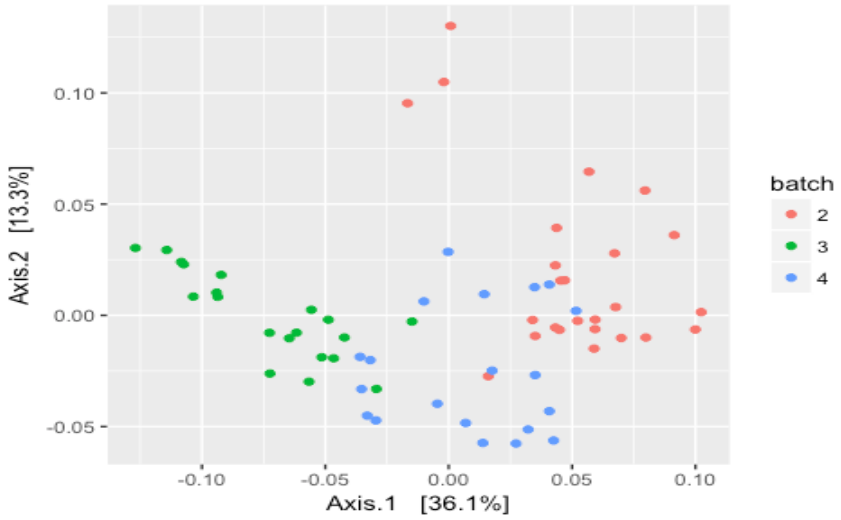
## Warning in estimate_richness(ps): The data you have provided does not have
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## I recommended that you find the un-trimmed data and retry.

## [1] 64.2

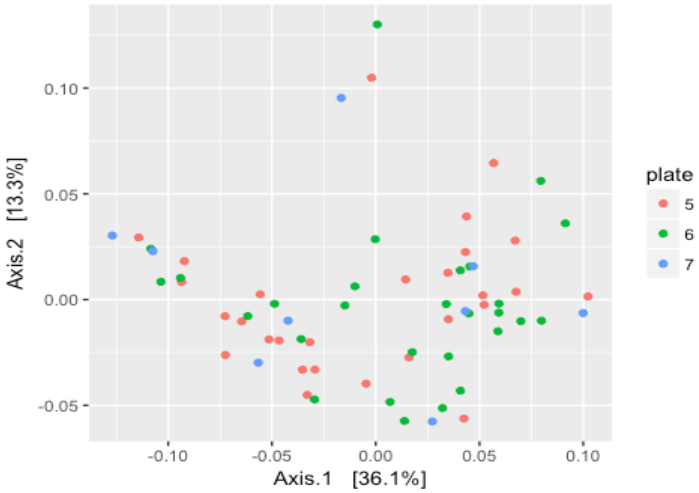
#Also, preprocess to remove taxa not observed at least once in 20% of samples. This is
good for beta diversity analyses since major drives in ecoystem diversity are often dri
ven by more abundant taxa. For alpha diversity and differential abundance analyses, I c
an use the raw count table
psf = filter_taxa(ps, function(x) sum(x > 1) > (0.20*length(x)), TRUE)

#Check if batch and plate may be affecting beta diversity
pslog <- transform_sample_counts(ps, function(x) log(1 + x))
out.wuf.log <- ordinate(pslog, method = "PCoA", distance = "wunifrac")

evals <- out.wuf.log$values$Eigenvalues
plot_ordination(pslog, out.wuf.log, color = "batch")
```




```r
plot_ordination(pslog, out.wuf.log, color = "plate")
```



YES, it appears that there is a batch effect on beta diversity. Plate, on the other hand, shows none. Use a batch effect correction after stabilizing for variance to correct for the effect. Put a number on how strong the effect is:

```
batch = get_variable(pslog, "batch")
batch_ano = anosim(phyloseq::distance(pslog,"wunifrac"),batch)

## Warning in UniFrac(physeq, weighted = TRUE, ...): Randomly assigning root
## as -- sq1223 -- in the phylogenetic tree in the data you provided.

summary(batch_ano)

##
## Call:
## anosim(dat = phyloseq::distance(pslog, "wunifrac"), grouping = batch)
## Dissimilarity:
##
## ANOSIM statistic R: 0.5707
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0342 0.0477 0.0624 0.0760
##
## Dissimilarity ranks between and within classes:
##         0%    25%    50%     75% 100%    N
## Between 29 797.50 1299.0 1707.25 2080 1400
## 2        1  99.75  361.0 1004.00 2050  300
## 3       40 404.25  628.5  886.75 1516  190
## 4       11 306.50  605.0  955.25 1683  190
```

ANOSIM stat of 0 means they are not similar to one another compared to random grouping, >1 means more similar. There is a substantial batch effect. Correct for it.

First transform phyloseq table to DESeq2 object, then assign batch numbers and perform variance stabilization. Before correcting for batch effect, plot PCA accounting for batch

```
ps_dds <- phyloseq_to_deseq2(psf, ~ treatment)

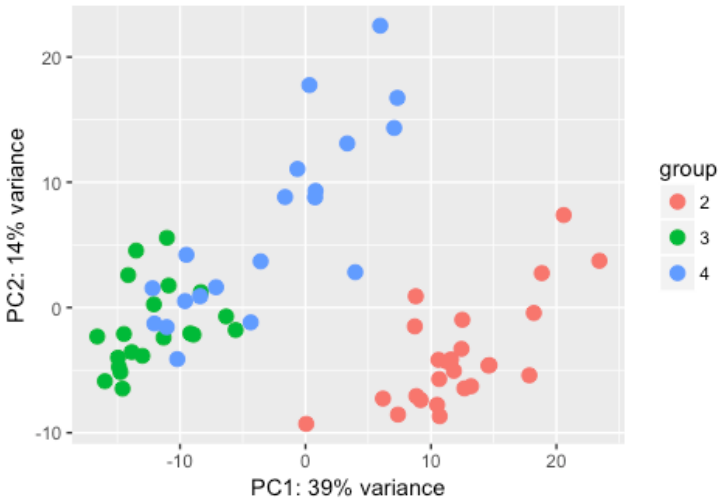
## Loading required namespace: DESeq2

## converting counts to integer mode
```

```
ps_dds$batch <- factor(as.data.frame(sample_data(psf))$batch)

vsd <- DESeq2::varianceStabilizingTransformation(ps_dds, blind = TRUE, fitType = "param
etric")

#plot to check without batch effect removed
DESeq2::plotPCA(vsd, "batch")
```

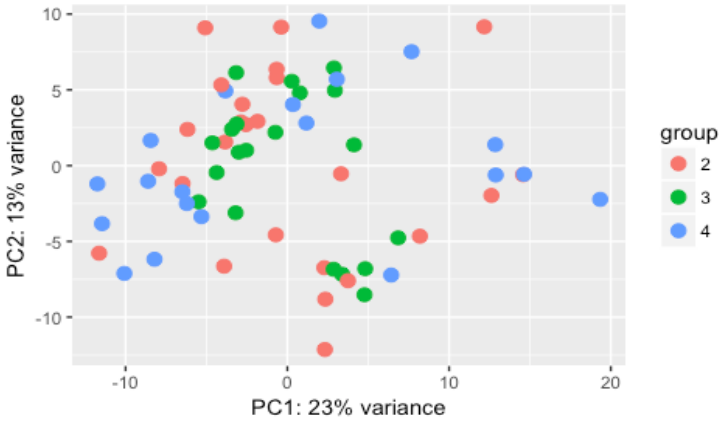


Then correct for batch effect and integrate this adjust table back into a phyloseq object. From limma documentation "The function (in effect) fits a linear model to the data, including both batches and regular treatments, then removes the component due to the batch effects."

```
#correct for batch effect
SummarizedExperiment::assay(vsd) <- limma::removeBatchEffect(SummarizedExperiment::assa
y(vsd), vsd$batch)

psrsvtab <- SummarizedExperiment::assay(vsd)

#check pca after batch effect corrected for
DESeq2::plotPCA(vsd, "batch")
```




```
ps.nobeffect <- psf
otu_table(ps.nobeffect) <- otu_table(psrsvtab, taxa_are_rows = TRUE)
```

Now I check to see if batch effect has at least decreased.

```
batch = get_variable(ps.nobeffect, "batch")
batch_ano = anosim(phyloseq::distance(ps.nobeffect,"wunifrac"),batch)
summary(batch_ano)

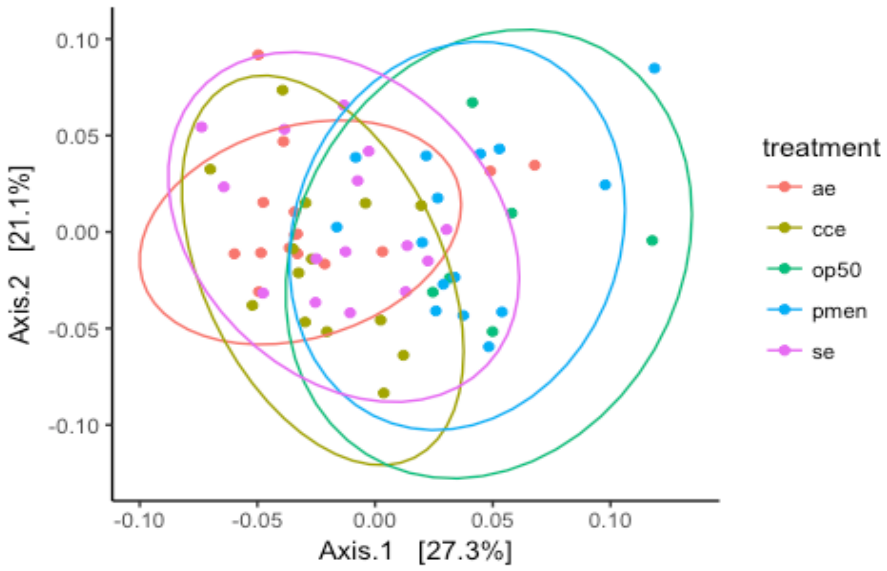
##
## Call:
## anosim(dat = phyloseq::distance(ps.nobeffect, "wunifrac"), grouping = batch)
## Dissimilarity:
##
## ANOSIM statistic R: 0.08323
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0268 0.0367 0.0452 0.0576
##
## Dissimilarity ranks between and within classes:
##         0%    25%    50%     75% 100%    N
## Between  9 573.75 1077.0 1567.25 2080 1400
## 2        2 452.50  957.5 1490.25 2077  300
## 3        1 280.25  703.5 1273.50 1976  190
## 4       16 594.50 1309.5 1784.00 2078  190
```

It has substantially decreased, from R2 ~ 0.55 to R2 ~ 0.08. Great!

Beta diversity

Run beta diversity analyses, first with plotting then to check if treatment has a significant effect.

```
ord.pc.un <- ordinate(ps.nobeffect, method = "PCoA", distance = "wunifrac")
evals <- ord.pc.un$values$Eigenvalues
plot_ordination(ps.nobeffect, ord.pc.un, color = "treatment") + stat_ellipse(type = "t"
) + theme_classic() + coord_fixed(sqrt(evals[2] / evals[1]))
```



Run ANOSIM to see if treatment has a significant effect on weighted unifrac grouping

```
treatment = get_variable(ps.nobeffect, "treatment")
treatment_ano = anosim(phyloseq::distance(ps.nobeffect,"wunifrac"),treatment)
summary(treatment_ano)

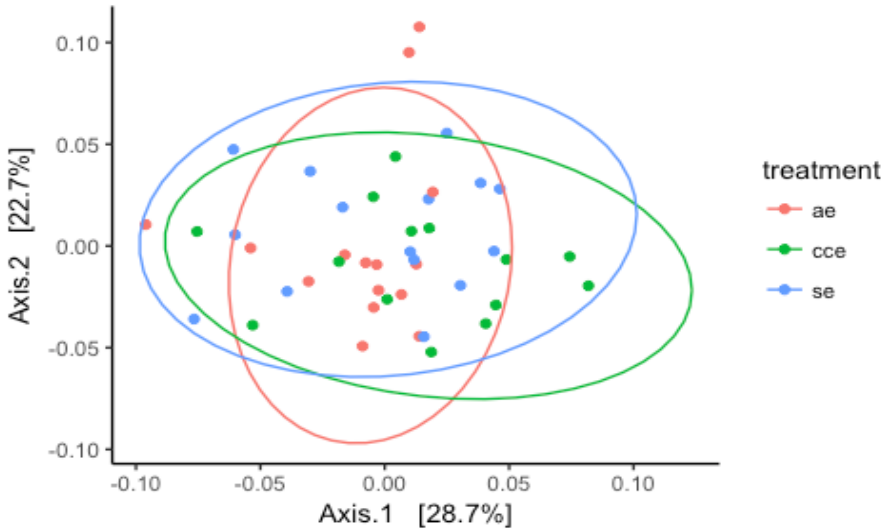
##
## Call:
## anosim(dat = phyloseq::distance(ps.nobeffect, "wunifrac"), grouping = treatment)
## Dissimilarity:
##
## ANOSIM statistic R: 0.201
##       Significance: 0.001
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0390 0.0515 0.0630 0.0753
##
## Dissimilarity ranks between and within classes:
##          0%   25%  50%    75% 100%    N
## Between   1 580.5 1088 1601.5 2080 1659
## ae        4 158.0  414 1444.0 1933  105
## cce       5 357.0  814 1162.0 2003   91
## op50    114 729.5 1359 1620.0 1952   15
## pmen      6 409.0 1167 1538.0 2036  105
## se       39 463.0  837 1346.0 1977  105
```

Yes, there is a small but significant effect on treatment on weighted unifrac groupings (R2 = 0.21, P < 0.01). Treatment here works as a significant but weak predictor of C. elegans microbiomes.

I can also show that there is no difference in microbiotas when subsetting to Enterococcus.

```
ps.nobeffect.Ef <- subset_samples(ps.nobeffect, treatment != "op50")
ps.nobeffect.Ef <- subset_samples(ps.nobeffect.Ef, treatment != "pmen")
```

```
ord.pc.un <- ordinate(ps.nobeffect.Ef, method = "PCoA", distance = "wunifrac")
evals <- ord.pc.un$values$Eigenvalues
plot_ordination(ps.nobeffect.Ef, ord.pc.un, color = "treatment") + stat_ellipse(type =
"t") +
    theme_classic() + coord_fixed(sqrt(evals[2]/evals[1]))
```




```
treatment = get_variable(ps.nobeffect.Ef, "treatment")
treatment_ano = anosim(phyloseq::distance(ps.nobeffect.Ef, "wunifrac"), treatment)
summary(treatment_ano)

##
## Call:
## anosim(dat = phyloseq::distance(ps.nobeffect.Ef, "wunifrac"),      grouping = treatm
ent)
## Dissimilarity:
##
## ANOSIM statistic R: 0.006402
##       Significance: 0.362
##
## Permutation: free
## Number of permutations: 999
##
## Upper quantiles of permutations (null model):
##    90%    95%  97.5%    99%
## 0.0347 0.0483 0.0574 0.0673
##
## Dissimilarity ranks between and within classes:
##         0% 25% 50% 75% 100%   N
## Between  1 242 471 707  946 645
## ae       3 128 307 788  933 105
## cce      4 276 528 678  943  91
## se      32 338 539 748  938 105
```

Run tests to see if treatment is significant for each, then adjust pvalues.

```
psb2 <- subset_samples(ps.nobeffect, batch == "2")
psb3 <- subset_samples(ps.nobeffect, batch == "3")
```

```
psb4 <- subset_samples(ps.nobeffect, batch == "4")

b2_treat_ando = anosim(phyloseq::distance(psb2, "uniFrac"), treatment)
b3_treat_ando = anosim(phyloseq::distance(psb3, "uniFrac"), treatment)
b4_treat_ando = anosim(phyloseq::distance(psb4, "uniFrac"), treatment)

# report mean and standard error of R2 for test
teststats = c(b2_treat_ando$statistic, b3_treat_ando$statistic, b4_treat_ando$statistic
)
mean(teststats)

## [1] -0.04153754

se <- function(x) sd(x)/sqrt(length(x))
se(teststats)

## [1] 0.04288196

bs_anosimps <- c(b2_treat_ando$signif, b3_treat_ando$signif, b4_treat_ando$signif)
p.adjust(bs_anosimps, method = "bonferroni", n = length(bs_anosimps))

## [1] 1.000 1.000 0.927
```

When treating batches individually, our results did not approach marginal significance that treatment had a significant effect on clustering (mean R2 (), se (), adj-Ps > 0.4). This isn't necessary to report but does show that I gained power from using multiple batches.

Alpha diversity

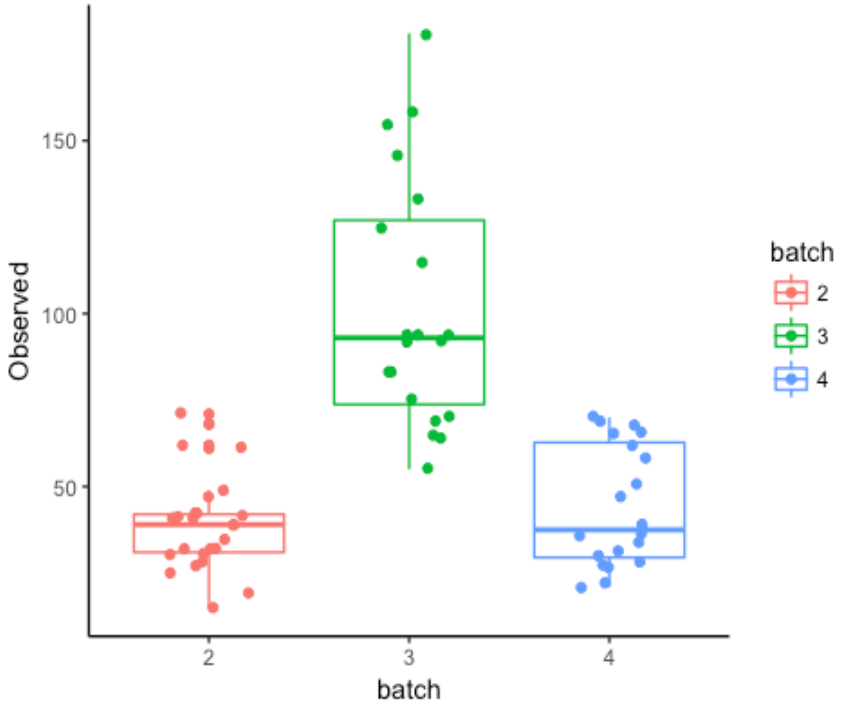
Run alpha diversity analyses on rarefied but unfiltered table, since difference in library size can affect results but also can singletons and doubletons and other lowly abundant bacteria.

```
psrare <- rarefy_even_depth(ps, min(sample_sums(ps)))

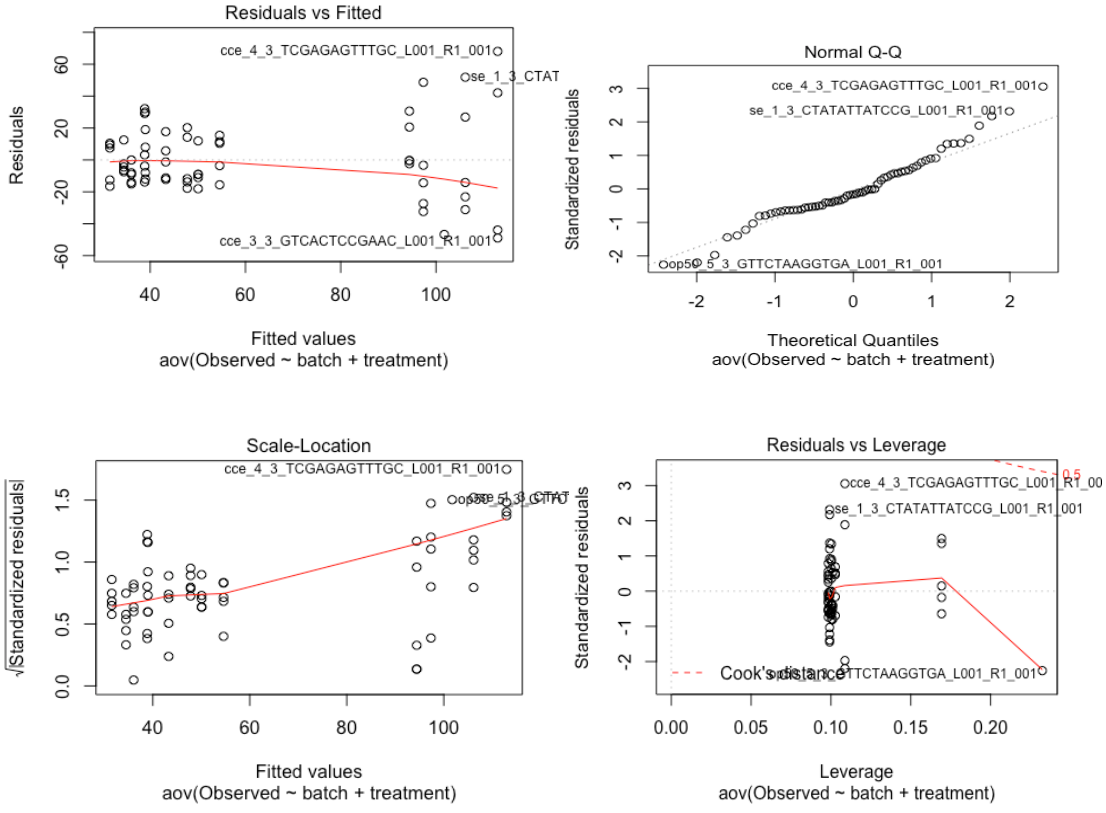
ps_richness <- estimate_richness(psrare)
# add sample data
ps_richness <- cbind(as.data.frame(sample_data(psrare)), ps_richness)
```

Using the Observed metric and a metric that incorporates evenness, I plot to see if batch is also affecting alpha diversity.

```
pO <- ggplot(ps_richness, aes(batch, Observed, color = batch))
pO <- pO + geom_boxplot(fill = NA)
pO + geom_point(position = position_jitter(width = 0.2)) + theme_classic()
```




```
sbt.obs <- (aov(Observed ~ batch + treatment, ps_richness))
plot(sbt.obs)
```



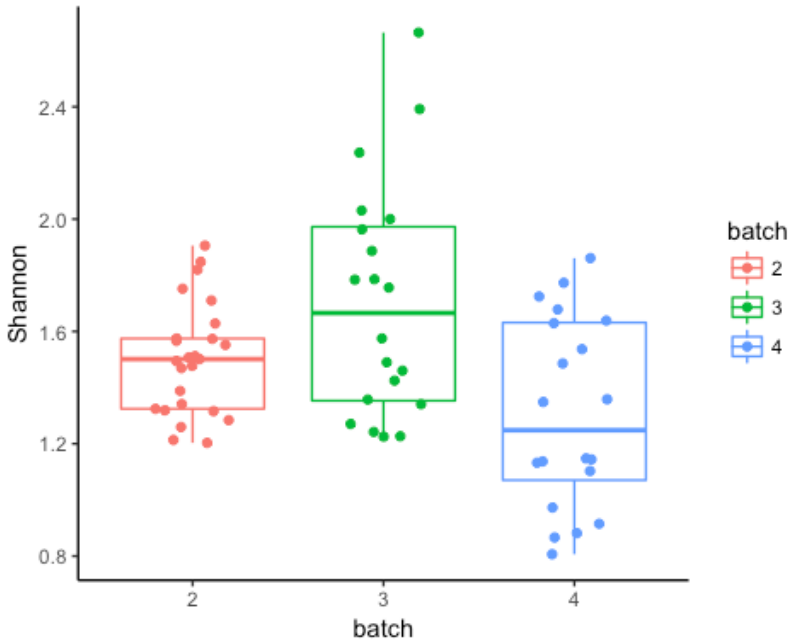

```
summary(sbt.obs)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## batch        2  50787   25394  45.583 1.27e-12 ***
## treatment    4   3106     776   1.394    0.247
## Residuals   58  32311     557
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

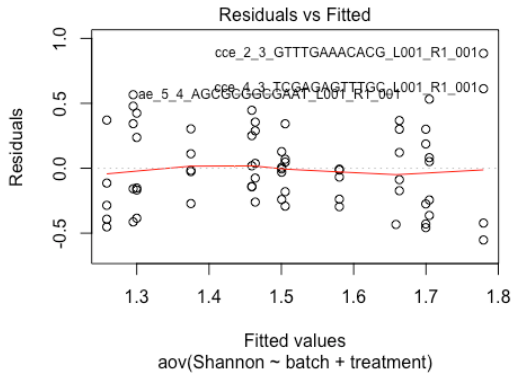
```
TukeyHSD(sbt.obs)
```

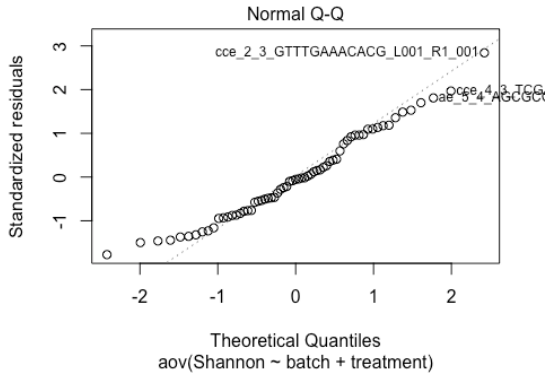
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Observed ~ batch + treatment, data = ps_richness)
##
## $batch
##       diff       lwr       upr     p adj
## 3-2  62.51  45.47840  79.54160 0.0000000
## 4-2   4.71 -12.32160  21.74160 0.7845145
## 4-3 -57.80 -75.75288 -39.84712 0.0000000
##
## $treatment
##                  diff        lwr       upr     p adj
## cce-ae     15.574048  -9.120336 40.268431 0.3974232
## op50-ae     4.388333 -27.711090 36.487757 0.9952199
## pmen-ae    -2.933333 -27.198217 21.331550 0.9970306
## se-ae       8.800000 -15.464883 33.064883 0.8446395
## op50-cce  -11.185714 -43.611029 21.239600 0.8669677
## pmen-cce  -18.507381 -43.201765  6.187003 0.2298197
## se-cce     -6.774048 -31.468431 17.920336 0.9375480
## pmen-op50  -7.321667 -39.421090 24.777757 0.9674189
## se-op50     4.411667 -27.687757 36.511090 0.9951212
## se-pmen    11.733333 -12.531550 35.998217 0.6543595
```

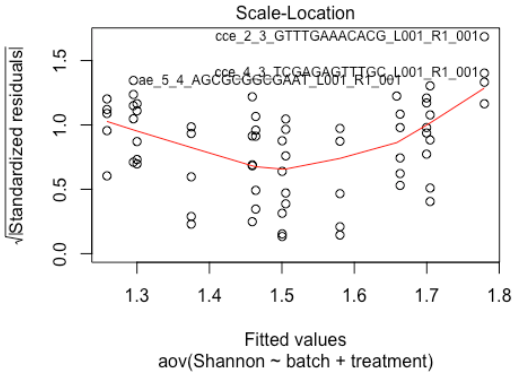
```
pS <- ggplot(ps_richness, aes(batch, Shannon, color = batch))
pS <- pS + geom_boxplot(fill = NA)
pS + geom_point(position = position_jitter(width = 0.2)) + theme_classic()
```

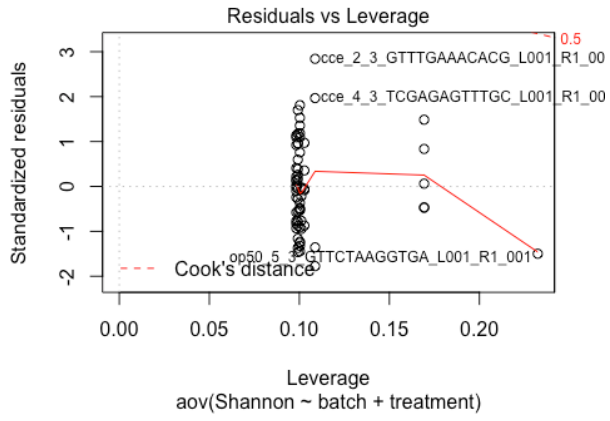



```
sbt.sdiv <- (aov(Shannon ~ batch + treatment, ps_richness))
plot(sbt.sdiv)
```










```
summary(sbt.sdiv)

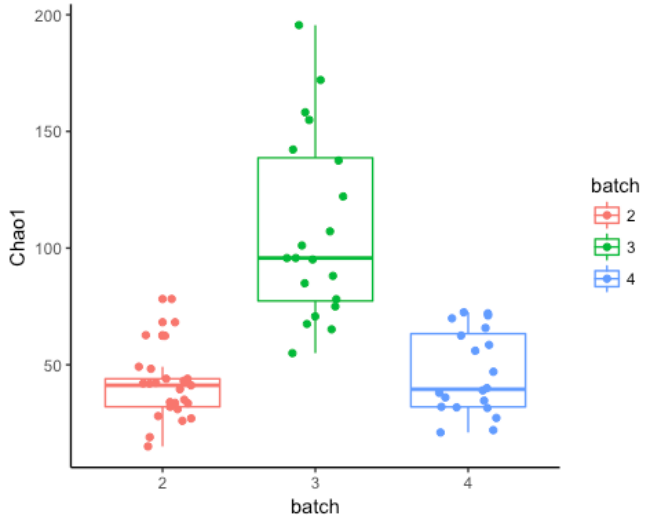
##             Df Sum Sq Mean Sq F value  Pr(>F)
## batch        2  1.589  0.7945   7.305 0.00148 **
## treatment    4  0.116  0.0290   0.267 0.89818
## Residuals   58  6.308  0.1088
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(sbt.sdiv)

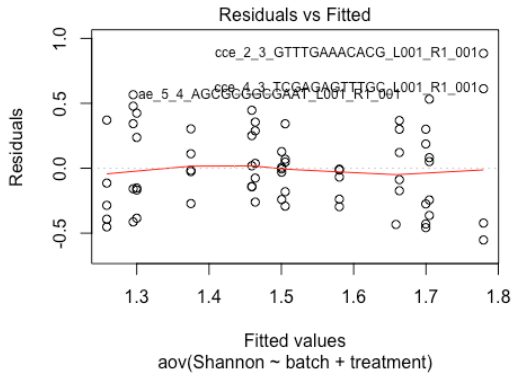
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Shannon ~ batch + treatment, data = ps_richness)
##
## $batch
##           diff         lwr         upr     p adj
## 3-2  0.2039285 -0.03404679  0.44190381 0.1070048
## 4-2 -0.1946554 -0.43263067  0.04331993 0.1294643
## 4-3 -0.3985839 -0.64943188 -0.14773589 0.0009369
##
## $treatment
##                    diff        lwr       upr     p adj
## cce-ae      0.079751189 -0.2652930 0.4247953 0.9658286
## op50-ae    -0.036943969 -0.4854556 0.4115677 0.9993400
## pmen-ae    -0.036517430 -0.3755604 0.3025255 0.9981066
## se-ae       0.004903861 -0.3341391 0.3439468 0.9999994
## op50-cce   -0.116695157 -0.5697603 0.3363700 0.9498122
```

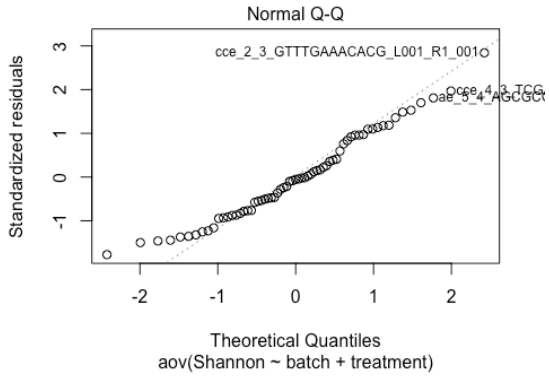
```
## pmen-cce  -0.116268618 -0.4613128 0.2287755 0.8764752
## se-cce    -0.074847327 -0.4198915 0.2701968 0.9728394
## pmen-op50  0.000426539 -0.4480851 0.4489382 1.0000000
## se-op50    0.041847830 -0.4066638 0.4903595 0.9989216
## se-pmen    0.041421291 -0.2976216 0.3804642 0.9969063

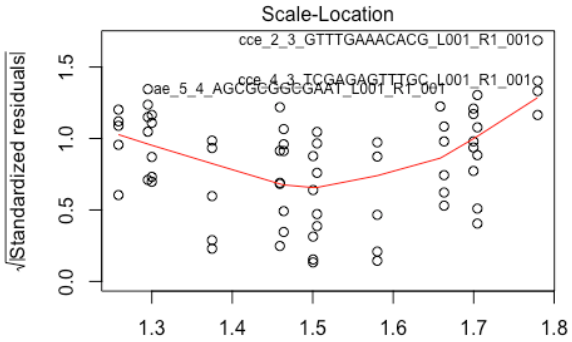
pC <- ggplot(ps_richness, aes(batch, Chao1, color = batch))
pC <- pC + geom_boxplot(fill = NA)
pC + geom_point(position = position_jitter(width = 0.2)) + theme_classic()
```

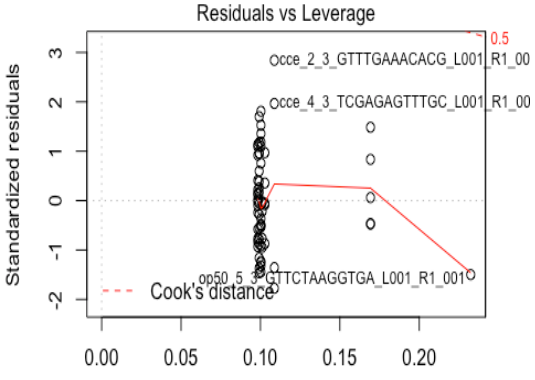



```
sbt.sdiv <- (aov(Shannon ~ batch + treatment, ps_richness))
plot(sbt.sdiv)
```










```
summary(sbt.sdiv)

##             Df Sum Sq Mean Sq F value  Pr(>F)
## batch        2  1.589  0.7945   7.305 0.00148 **
## treatment    4  0.116  0.0290   0.267 0.89818
## Residuals   58  6.308  0.1088
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(sbt.sdiv)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Shannon ~ batch + treatment, data = ps_richness)
##
## $batch
##           diff         lwr         upr     p adj
## 3-2  0.2039285 -0.03404679  0.44190381 0.1070048
## 4-2 -0.1946554 -0.43263067  0.04331993 0.1294643
## 4-3 -0.3985839 -0.64943188 -0.14773589 0.0009369
##
## $treatment
##                   diff        lwr       upr     p adj
## cce-ae     0.079751189 -0.2652930 0.4247953 0.9658286
## op50-ae   -0.036943969 -0.4854556 0.4115677 0.9993400
## pmen-ae   -0.036517430 -0.3755604 0.3025255 0.9981066
## se-ae      0.004903861 -0.3341391 0.3439468 0.9999994
```

```
## op50-cce  -0.116695157 -0.5697603 0.3363700 0.9498122
## pmen-cce  -0.116268618 -0.4613128 0.2287755 0.8764752
## se-cce    -0.074847327 -0.4198915 0.2701968 0.9728394
## pmen-op50  0.000426539 -0.4480851 0.4489382 1.0000000
## se-op50    0.041847830 -0.4066638 0.4903595 0.9989216
## se-pmen    0.041421291 -0.2976216 0.3804642 0.9969063
```

Diversity is higher in batch 3 than the other two, a result which is significant. I remove batch 3 for alpha diversity measurements.

```
psn3 <- subset_samples(ps, batch != "3")
psn3rare <- rarefy_even_depth(psn3, min(sample_sums(psn3)))

## You set `rngseed` to FALSE. Make sure you've set & recorded
##  the random seed of your session for reproducibility.
## See `?set.seed`

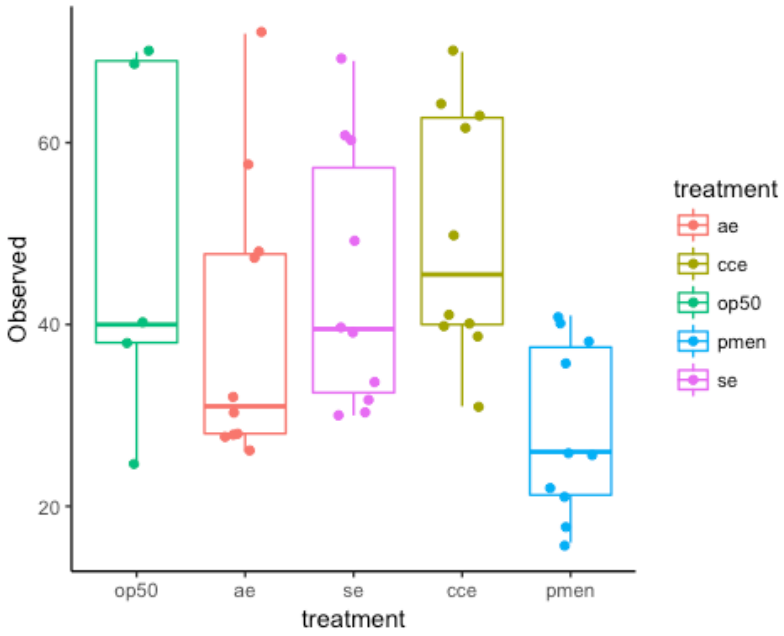
## ...

## 975OTUs were removed because they are no longer
## present in any sample after random subsampling

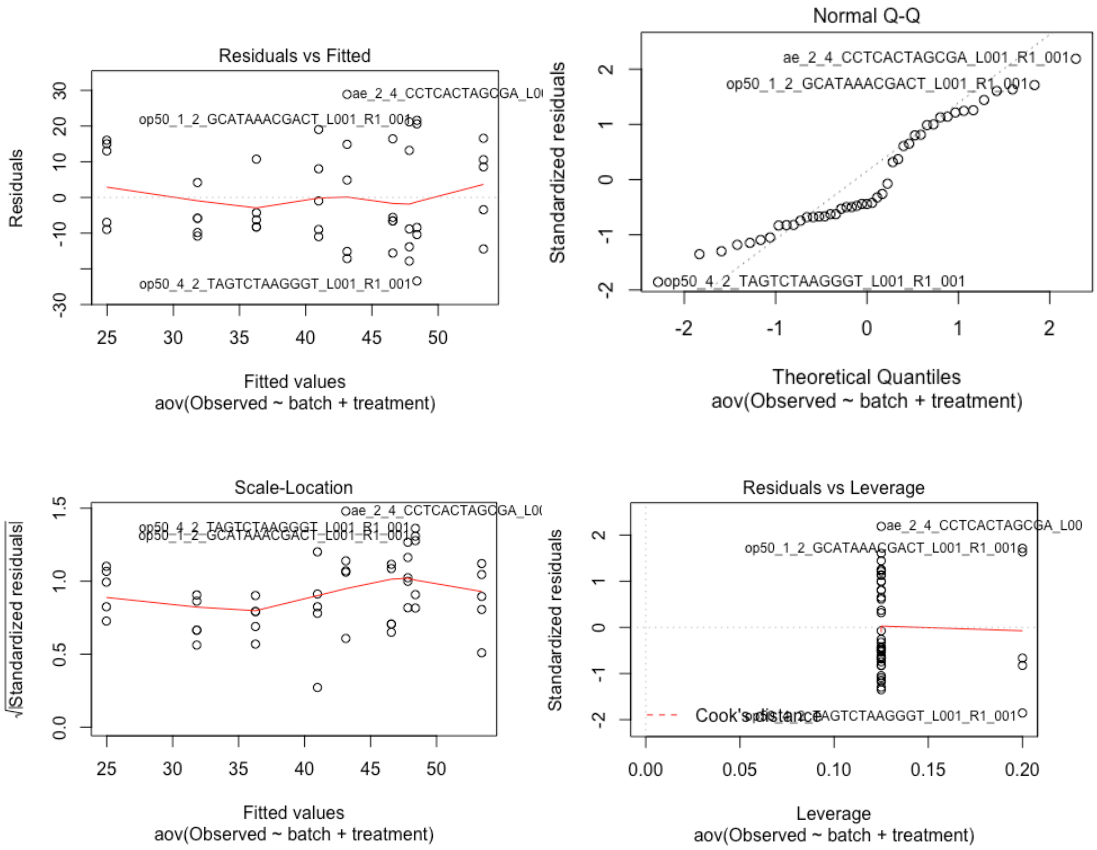
## ...

psn3_richness <- estimate_richness(psn3rare)
psn3_richness <- cbind(as.data.frame(sample_data(psn3rare)), psn3_richness)

pO2 <- ggplot(psn3_richness, aes(treatment, Observed, color = treatment))
pO2 <- pO2 + geom_boxplot(fill = NA)
pO2 + geom_point(position = position_jitter(width = 0.2)) + theme_classic() +
    scale_x_discrete(limits = c("op50", "ae", "se", "cce", "pmen"))
```




```
sbt.obs2 <- (aov(Observed ~ batch + treatment, psn3_richness))
plot(sbt.obs2)
```




```
summary(sbt.obs2)

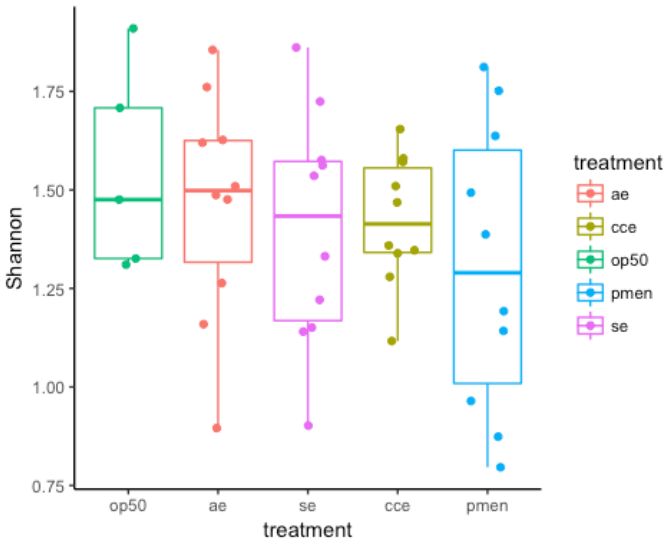
##             Df Sum Sq Mean Sq F value Pr(>F)
## batch        1    236   236.1   1.188 0.2825
## treatment    4   3026   756.6   3.805 0.0105 *
## Residuals   39   7755   198.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(sbt.obs2)

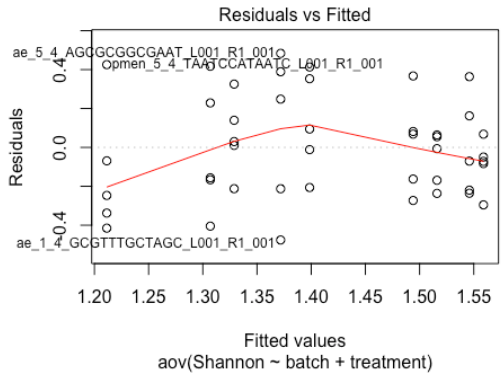
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Observed ~ batch + treatment, data = psn3_richness)
##
## $batch
##     diff       lwr      upr     p adj
## 4-2 4.61 -3.946692 13.16669 0.2825136
##
## $treatment
##              diff        lwr        upr     p adj
## cce-ae     10.300  -7.732576 28.3325757 0.4859036
## op50-ae    11.005 -11.080305 33.0903046 0.6157890
## pmen-ae   -11.300 -29.332576  6.7325757 0.3926795
## se-ae       4.700 -13.332576 22.7325757 0.9443130
## op50-cce    0.705 -21.380305 22.7903046 0.9999835
## pmen-cce  -21.600 -39.632576 -3.5674243 0.0119956
## se-cce     -5.600 -23.632576 12.4325757 0.8996130
```

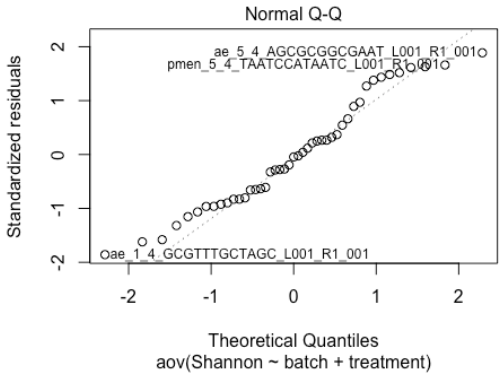
```
## pmen-op50 -22.305 -44.390305 -0.2196954 0.0467471
## se-op50    -6.305 -28.390305 15.7803046 0.9240173
## se-pmen    16.000  -2.032576 34.0325757 0.1030241

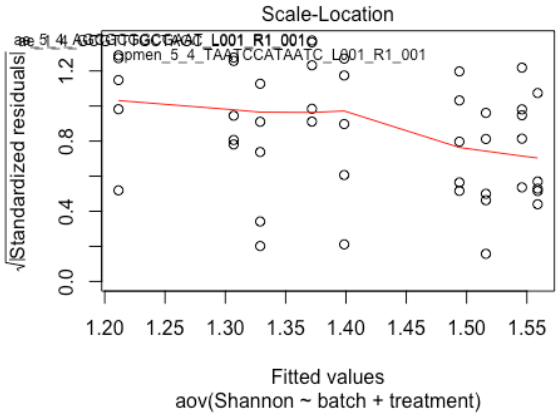
pS2 <- ggplot(psn3_richness, aes(treatment, Shannon, color = treatment))
pS2 <- pS2 + geom_boxplot(fill = NA)
pS2 + geom_point(position = position_jitter(width = 0.2)) + theme_classic() +
    scale_x_discrete(limits = c("op50", "ae", "se", "cce", "pmen"))
```

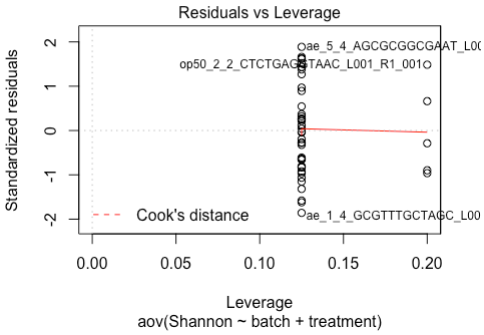



```
sbt.shan2 <- (aov(Shannon ~ batch + treatment, psn3_richness))
plot(sbt.shan2)
```










```
summary(sbt.shan2)

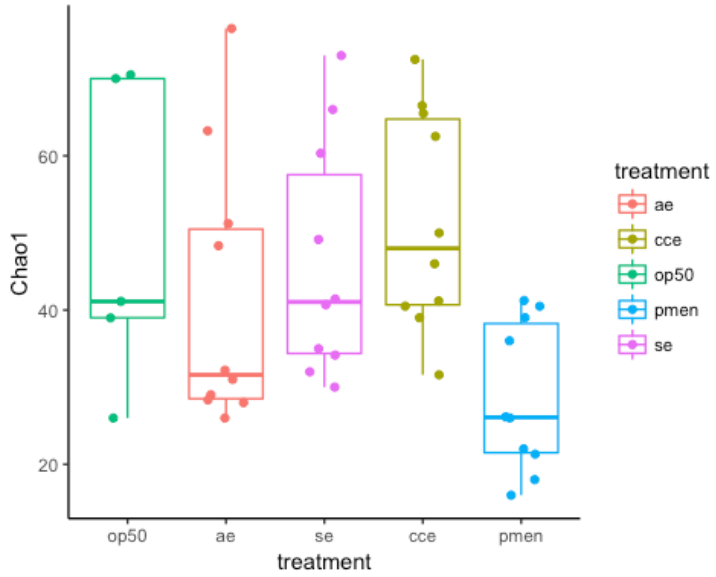
##             Df Sum Sq Mean Sq F value Pr(>F)
## batch        1 0.4353  0.4353   5.803 0.0208 *
## treatment    4 0.1496  0.0374   0.499 0.7368
## Residuals   39 2.9256  0.0750
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(sbt.shan2)

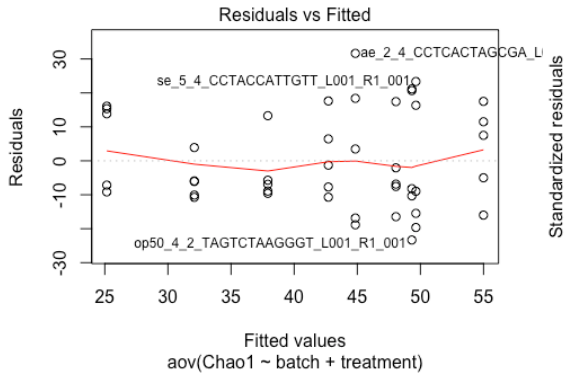
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Shannon ~ batch + treatment, data = psn3_richness)
##
## $batch
##           diff        lwr         upr     p adj
## 4-2 -0.1979393 -0.3641374 -0.03174128 0.0208183
##
## $treatment
##                  diff        lwr       upr     p adj
## cce-ae    -0.04290272 -0.3931525 0.3073471 0.9966329
## op50-ae   -0.01842950 -0.4473961 0.4105371 0.9999462
## pmen-ae   -0.16039345 -0.5106432 0.1898563 0.6870328
## se-ae     -0.06488560 -0.4151354 0.2853642 0.9837305
## op50-cce   0.02447322 -0.4044934 0.4534398 0.9998336
## pmen-cce  -0.11749072 -0.4677405 0.2327591 0.8715122
## se-cce    -0.02198287 -0.3722327 0.3282669 0.9997569
```

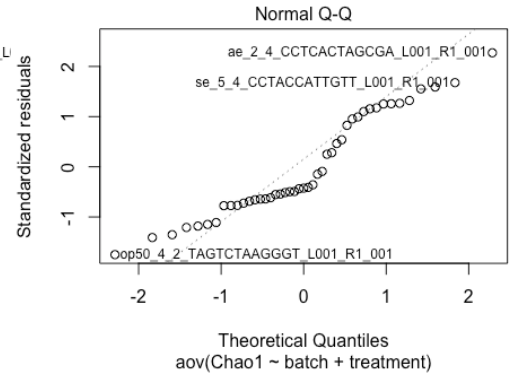
```
## pmen-op50 -0.14196395 -0.5709306 0.2870027 0.8768742
## se-op50   -0.04645610 -0.4754227 0.3825105 0.9979145
## se-pmen    0.09550785 -0.2547419 0.4457576 0.9349546

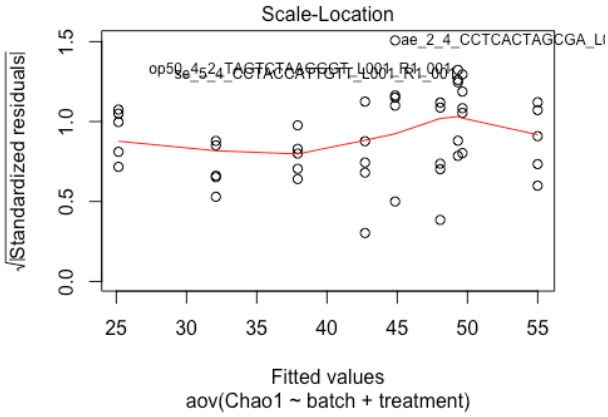
pC2 <- ggplot(psn3_richness, aes(treatment, Chao1, color = treatment))
pC2 <- pC2 + geom_boxplot(fill = NA)
pC2 + geom_point(position = position_jitter(width = 0.2)) + theme_classic() +
    scale_x_discrete(limits = c("op50", "ae", "se", "cce", "pmen"))
```

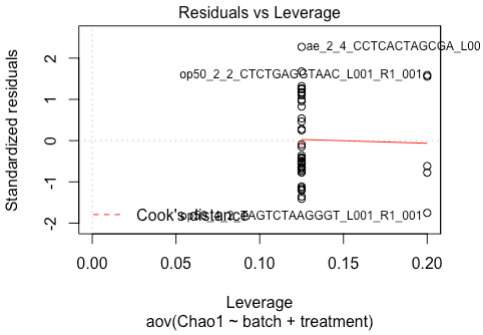



```
sbt.chao12 <- (aov(Chao1 ~ batch + treatment, psn3_richness))
plot(sbt.chao12)
```










```
summary(sbt.chao12)

##             Df Sum Sq Mean Sq F value Pr(>F)
## batch        1    252   252.1   1.137 0.2929
## treatment    4   3347   836.8   3.774 0.0109 *
## Residuals   39   8647   221.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(sbt.chao12)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Chao1 ~ batch + treatment, data = psn3_richness)
##
## $batch
##         diff       lwr     upr     p adj
## 4-2 4.762913 -4.272375 13.7982 0.2928659
##
## $treatment
##                  diff        lwr        upr     p adj
## cce-ae     10.1483333  -8.892847 29.1895140 0.5536028
## op50-ae    10.3220119 -12.998577 33.6426003 0.7132809
## pmen-ae   -12.7566667 -31.797847  6.2845140 0.3262884
## se-ae       4.7945238 -14.246657 23.8357045 0.9506132
## op50-cce    0.1736786 -23.146910 23.4942670 1.0000000
## pmen-cce  -22.9050000 -41.946181 -3.8638193 0.0115375
## se-cce     -5.3538095 -24.394990 13.6873712 0.9278187
```

```
## pmen-op50 -23.0786786 -46.399267  0.2419098 0.0536029
## se-op50    -5.5274881 -28.848077 17.7931003 0.9600907
## se-pmen    17.5511905  -1.489990 36.5923712 0.0832562
```

Treatment does not affect alpha diversity measurements strongly. The only observed significant effect is that P. mendocina has significantly fewer observed species than CCE.

Make a table reporting mean and se differences in alpha diversity using main metrics, figures are provided as supplementary.

```
psn3_richness_mean <- ddply(psn3_richness, .(treatment), summarize, Observed = mean(Obs
erved),
    Shannon = mean(Shannon), Chao1 = mean(Chao1))

psn3_richness_se <- ddply(psn3_richness, .(treatment), summarize, se.Observed = se(Obse
rved),
    se.Shannon = se(Shannon), se.Chao1 = se(Chao1))

psn3_richness_summary <- cbind(psn3_richness_mean, psn3_richness_se)

write.csv(psn3_richness_summary, "~/Documents/King_Lab/Masters_thesis/tables/psn3_richn
ess.csv")

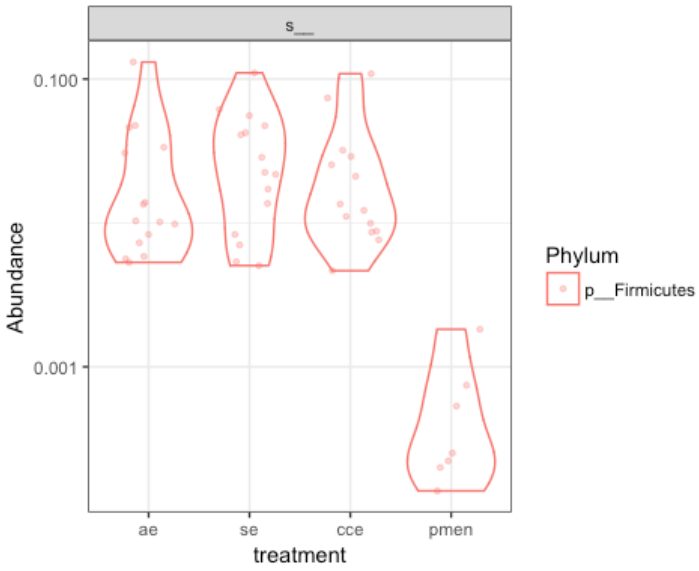
plot_abundance = function(ps, title = "", Facet = "Order", Color = "Phylum") {
    # Arbitrary subset, based on Phylum, for plotting
    p1f = subset_taxa(ps, Kingdom %in% c("k__Bacteria"))
    mphyseq = psmelt(p1f)
    mphyseq <- subset(mphyseq, Abundance > 0)
    ggplot(data = mphyseq, mapping = aes_string(x = "treatment", y = "Abundance",
        color = Color, fill = Color)) + geom_violin(fill = NA) + geom_point(size = 1,
        alpha = 0.3, position = position_jitter(width = 0.3)) + facet_wrap(facets = Fac
et) +
        scale_y_log10() + theme(legend.position = "none")
}

psr = transform_sample_counts(ps, function(x) (x/sum(x)))

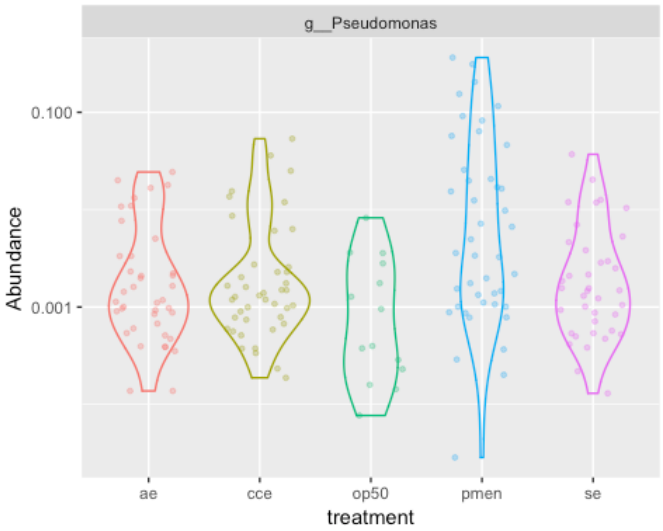
psEf = subset_taxa(psr, Genus == "g__Enterococcus")

## Warning in prune_taxa(taxa, phy_tree(x)): prune_taxa attempted to reduce tree to 1 o
r fewer tips.
##  tree replaced with NULL.

plot_abundance(psEf, Facet = "Species") + theme_bw() + scale_x_discrete(limits = c("ae"
,
    "se", "cce", "pmen"))
```

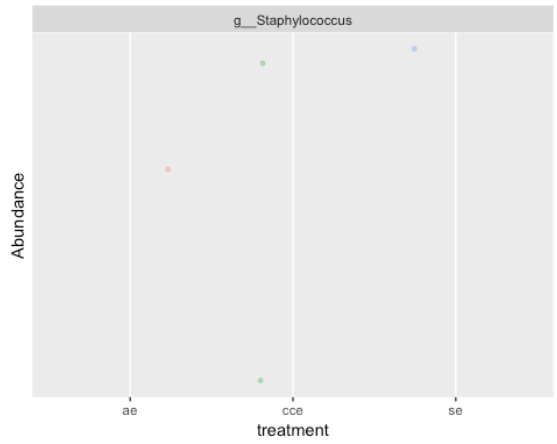



```
psPs = subset_taxa(psr, Genus == "g__Pseudomonas")
plot_abundance(psPs, Facet = "Genus", Color = "treatment")
```




```
# Staph is basically nonexistent in samples
psSt = subset_taxa(psr, Genus == "g__Staphylococcus")
plot_abundance(psSt, Facet = "Genus", Color = "treatment")

## Warning in max(data$density): no non-missing arguments to max; returning -
## Inf
```




```
# In soil?
pssoilsumm <- summarize_taxa(pssoil, "Genus", GroupBy = "sample.id")
taxsumm.sp <- subset(pssoilsumm, Genus == "g__Staphylococcus")
taxsumm.sp

## Empty data.table (0 rows) of 7 cols: Genus,sample.id,meanRA,sdRA,seRA,minRA...

# No
```

Enterococcus is observed primarily in Enterococcus treatments. It is not found at all in OP50 treatment, perhaps because it has no shared evolutioanry history with C. elegans it does not colonize unless preexposed. It is observed in the pmen treatment, which is particualrly interesting since Montalvo-Katz et al. previously showed that P. mendocina does not inhibit E. faecalis colonization.

Pseudomonas is found in all samples.

Nearly nothing for Staphylococcus. With preferential colonization by other soil microbes, this isn't surprising.

Differential abundance analysis

For the beta diversity analysis I created a transformed version of the data to account for batch effect in our multivariate analysis, now I differently include batch effect in our design formula when testing for differentially abundant RSVs due to treatment.

Most the code in this tutorial is taken from Callahan et al. 2016 workflow or the DESeq2 bioconductor tutorial.

First design the formula then run formal DESeq test then extract results with specific contrasts.

Want to know how Enterococcus differs from OP50, how Pmen differs from OP50, and how CCE differs from SE and AE.

```
# Make phyloseq object for deseq2
dds.all <- phyloseq_to_deseq2(ps, ~batch + treatment)

## converting counts to integer mode
```

Run DESeq test and extract results. Can extract log2fold change results from each with specific comparisons using the contrast argument.

```
dds.all <- DESeq2::DESeq(dds.all, test = "Wald", fitType = "parametric")

resMF.ae <- DESeq2::results(dds.all, contrast = c("treatment", "ae", "op50"))
resMF.se <- DESeq2::results(dds.all, contrast = c("treatment", "se", "op50"))
resMF.cce <- DESeq2::results(dds.all, contrast = c("treatment", "cce", "op50"))
```

```
resMF.pmen <- DESeq2::results(dds.all, contrast = c("treatment", "pmen", "op50"))

resMF.cce.ae <- DESeq2::results(dds.all, contrast = c("treatment", "cce", "ae"))
resMF.cce.se <- DESeq2::results(dds.all, contrast = c("treatment", "cce", "se"))
resMF.se.ae <- DESeq2::results(dds.all, contrast = c("treatment", "se", "ae"))
```

I now order by adjusted p-value, remove those with NA value and format the table and taxonomy for plotting. Plotting results is taken from [phyloseq to DESeq2 tutorial].
Start with broader comparison.

```
alpha = 0.05
theme_set(theme_bw())

resMF.ae.sig <- resMF.ae[which(resMF.ae$padj < alpha), ] %>% cbind(as(., "data.frame"),

    as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.ae.sig[, "treatment"] <- rep("ae", dim(resMF.ae.sig)[1])
resMF.ae.sig[, "sq"] <- row.names(resMF.ae.sig)

resMF.se.sig <- resMF.se[which(resMF.se$padj < alpha), ] %>% cbind(as(., "data.frame"),

    as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.se.sig[, "treatment"] <- rep("se", dim(resMF.se.sig)[1])
resMF.se.sig[, "sq"] <- row.names(resMF.se.sig)

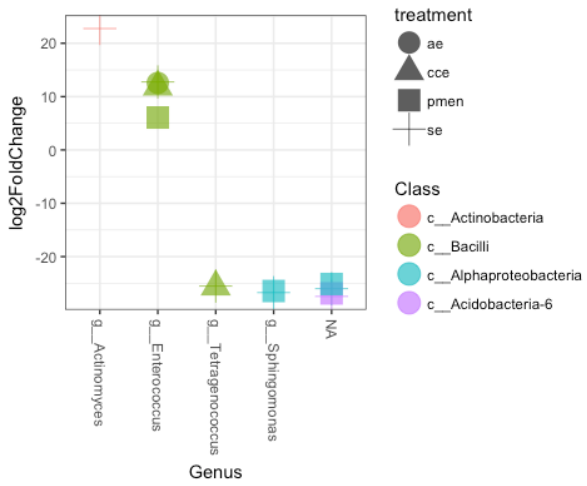
resMF.cce.sig <- resMF.cce[which(resMF.cce$padj < alpha), ] %>% cbind(as(.,
    "data.frame"), as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.cce.sig[, "treatment"] <- rep("cce", dim(resMF.cce.sig)[1])
resMF.cce.sig[, "sq"] <- row.names(resMF.cce.sig)


resMF.pmen.sig <- resMF.pmen[which(resMF.pmen$padj < alpha), ] %>% cbind(as(.,
    "data.frame"), as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.pmen.sig[, "treatment"] <- rep("pmen", dim(resMF.pmen.sig)[1])
resMF.pmen.sig[, "sq"] <- row.names(resMF.pmen.sig)


resMF.e.o.p.sig.tab <- rbind(resMF.ae.sig, resMF.se.sig, resMF.cce.sig, resMF.pmen.sig)

# Class order
x = tapply(resMF.e.o.p.sig.tab$log2FoldChange, resMF.e.o.p.sig.tab$Class, function(x) m
ax(x))
x = sort(x, TRUE)
resMF.e.o.p.sig.tab$Class = factor(as.character(resMF.e.o.p.sig.tab$Class),
    levels = names(x))
# Genus order
x = tapply(resMF.e.o.p.sig.tab$log2FoldChange, resMF.e.o.p.sig.tab$Genus, function(x) m
ax(x))
x = sort(x, TRUE)
resMF.e.o.p.sig.tab$Genus = factor(as.character(resMF.e.o.p.sig.tab$Genus),
    levels = names(x))
resMF.e.o.p.sig.tab$Genus <- replace(resMF.e.o.p.sig.tab$Genus, resMF.e.o.p.sig.tab$Gen
us ==
    "g__", NA)


ggplot(resMF.e.o.p.sig.tab, aes(x = Genus, y = log2FoldChange, color = Class,
    shape = treatment)) + geom_point(size = 6, alpha = 0.7) + theme(axis.text.x = eleme
nt_text(angle = -90,
    hjust = 0, vjust = 0.5))
```



Then compare CCE to AE and SE, and SE to AE

```
resMF.cce.ae.sig <- resMF.cce.ae[which(resMF.cce.ae$padj < alpha), ] %>% cbind(as(.,
    "data.frame"), as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.cce.ae.sig[, "treatment"] <- rep("CCE/AE", dim(resMF.cce.ae.sig)[1])
resMF.cce.ae.sig[, "sq"] <- row.names(resMF.cce.ae.sig)

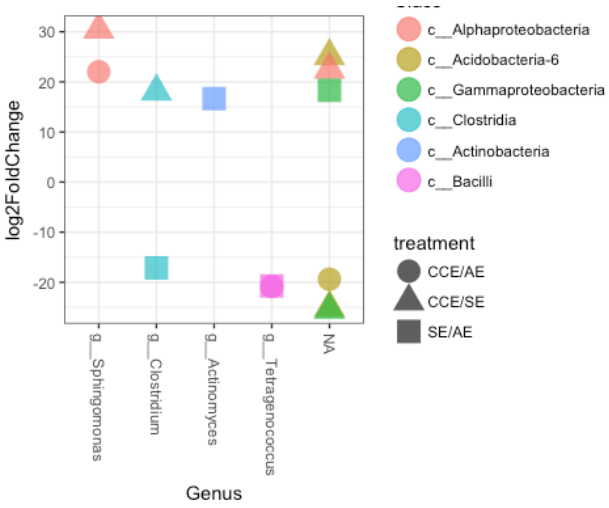

resMF.cce.se.sig <- resMF.cce.se[which(resMF.cce.se$padj < alpha), ] %>% cbind(as(.,
    "data.frame"), as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.cce.se.sig[, "treatment"] <- rep("CCE/SE", dim(resMF.cce.se.sig)[1])
resMF.cce.se.sig[, "sq"] <- row.names(resMF.cce.se.sig)


resMF.se.ae.sig <- resMF.se.ae[which(resMF.se.ae$padj < alpha), ] %>% cbind(as(.,
    "data.frame"), as(tax_table(ps)[rownames(.), ], "matrix")) %>% data.frame(.)
resMF.se.ae.sig[, "treatment"] <- rep("SE/AE", dim(resMF.se.ae.sig)[1])
resMF.se.ae.sig[, "sq"] <- row.names(resMF.se.ae.sig)


resMF.e.sig.tab <- rbind(resMF.cce.ae.sig, resMF.cce.se.sig, resMF.se.ae.sig)

# Class order
x = tapply(resMF.e.sig.tab$log2FoldChange, resMF.e.sig.tab$Class, function(x) max(x))
x = sort(x, TRUE)
resMF.e.sig.tab$Class = factor(as.character(resMF.e.sig.tab$Class), levels = names(x))
# Genus order
x = tapply(resMF.e.sig.tab$log2FoldChange, resMF.e.sig.tab$Genus, function(x) max(x))
x = sort(x, TRUE)
resMF.e.sig.tab$Genus = factor(as.character(resMF.e.sig.tab$Genus), levels = names(x))
resMF.e.sig.tab$Genus <- replace(resMF.e.sig.tab$Genus, resMF.e.sig.tab$Genus ==
    "g__", NA)


ggplot(resMF.e.sig.tab, aes(x = Genus, y = log2FoldChange, color = Class, shape = treat
ment)) +
    geom_point(size = 6, alpha = 0.7) + theme(axis.text.x = element_text(angle = -90,
    hjust = 0, vjust = 0.5))
```



Taxa agglomeration for gene change, colonization and protection correlations

Since resolution is limited using the 16S genes and I cannot distinguish between Enterococcus species or other species of interest, I agglomerate and summarize taxa at the genus level and use these values to draw correlations between transcript levels and colonization prior to soil exposure and to protection post soil exposure.

First define functions, taken from phyloseq github forum.

Pull mean and se for Enterococcus from samples

```
taxsumm <- summarize_taxa(ps, "Genus", GroupBy = "treatment")
taxsumm.e <- subset(taxsumm, Genus == "g__Enterococcus")
```

Load other data - colonization and protection data as well as TPM values.

```
# Colonization and survival data
colsurv <- read.csv(file = "~/Documents/King_Lab/Masters_thesis/gut_surv_data/colonize_
surv.csv")

colsurv.summ <- ddply(colsurv, .(treatment), summarize, mean.cfus = mean(cfus),
    se.cfus = se(cfus), mean.prop.dead = mean(prop.dead), se.prop.dead = se(prop.dead))

somxdf <- read.csv(file = "~/Documents/King_Lab/Masters_thesis/RNASeq/sleuth/somxdf.csv
")
# do the same combining but for transcripts that map to genes, clec-48 and
# ilys-3 as well as epithelial transcripts ZC449.1, ZC449.2, H03E18.1,
# H42K12.3, T26C5.2
somxdf.summ <- ddply(somxdf, .(condition), summarize, mean.clec48 = mean(C14A6.1),
    se.clec48 = se(C14A6.1), mean.ilys3 = mean(C45G7.3), se.ilys3 = se(C45G7.3),
    mean.B0024.4 = mean(B0024.4), se.B0024.4 = se(B0024.4), mean.Y47H9C.1 = mean(Y47H9C
.1),
    se.Y47H9C.1 = se(Y47H9C.1), mean.cnc6 = mean(Y46E12A.1), se.cnc6 = se(Y46E12A.1),
    mean.vhp1 = mean(F08B1.1c.2), se.vhp1 = se(F08B1.1c.2), mean.ilys3 = mean(C45G7.3),

    mean.ZC449.1 = mean(ZC449.1), mean.ZC449.2 = mean(ZC449.2), mean.H03E18.1 = mean(H0
3E18.1),
    mean.H42K12.3 = mean(H42K12.3.1), mean.T26C5.2 = mean(T26C5.2))

somxdf.summ[, "condition"] <- tolower(somxdf.summ$condition)
colnames(somxdf.summ)[1] <- "treatment"

somxdf.summ.epit <- somxdf.summ[, -which(names(somxdf.summ) %in% c("mean.clec48",
    "se.clec48", "mean.ilys3", "se.ilys3"))]
somxdf.summ.epit <- melt(somxdf.summ.epit, id.vars = "treatment")
```

```
# Need to confirm since it wouldn't make sense that SE has fewer than AE,
# since it decreased in expression

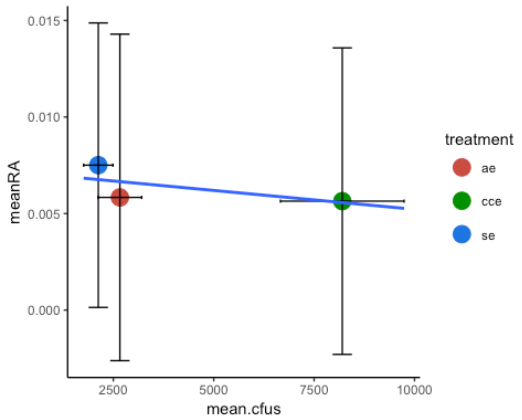
allsumms <- merge(taxsumm.e, colsurv.summ, by = "treatment") %>% merge(., somxdf.summ,
    by = "treatment")
somxdf.summ.epit[, "mean.cfus"] <- allsumms$mean.cfus

summ.cfus.lims <- aes(xmax = allsumms$mean.cfus + allsumms$se.cfus, xmin = allsumms$mea
n.cfus -
    allsumms$se.cfus)
summ.e.lims <- aes(ymax = allsumms$meanRA + allsumms$seRA, ymin = allsumms$meanRA -
    allsumms$seRA)

p3 <- ggplot(allsumms, aes(mean.cfus, meanRA)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 8) + geom_errorbar(summ.e.lim
s,
    position = "dodge") + geom_errorbarh(summ.cfus.lims, position = "dodge") +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE) + theme_classic()


p3

## Warning: position_dodge requires non-overlapping x intervals
```




```
# With mean RA as response and cfus prior to exposure as predictor
cor.test(allsumms$meanRA, allsumms$mean.cfus, method = "p")

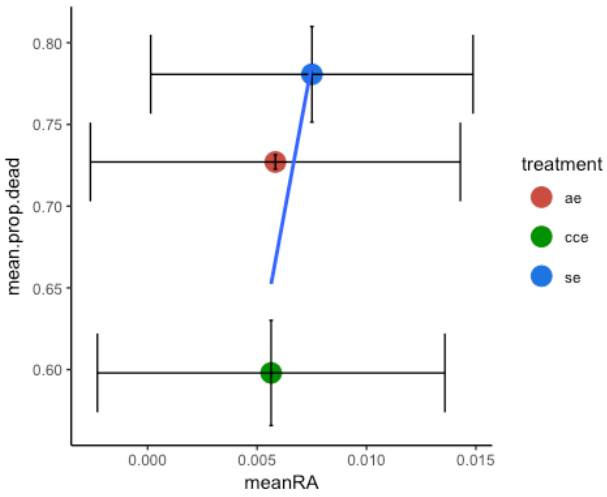
##
##  Pearson's product-moment correlation
##
## data:  allsumms$meanRA and allsumms$mean.cfus
## t = -0.83906, df = 1, p-value = 0.5556
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##       cor
## -0.642772

summ.prop.deadlims <- aes(xmax = allsumms$mean.prop.dead + allsumms$se.prop.dead,
    xmin = allsumms$mean.prop.dead - allsumms$se.prop.dead)
summ.prop.deadlims.y <- aes(ymax = allsumms$mean.prop.dead + allsumms$se.prop.dead,
    ymin = allsumms$mean.prop.dead - allsumms$se.prop.dead)
summ.e.lims.x <- aes(xmax = allsumms$meanRA + allsumms$seRA, xmin = allsumms$meanRA -
    allsumms$seRA)
```

```
p4 <- ggplot(allsumms, aes(meanRA, mean.prop.dead)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 8) + geom_errorbarh(summ.e.li
ms.x,
    position = "dodge") + geom_errorbar(summ.prop.deadlims.y, position = "dodge") +
    geom_smooth(method = lm, se = FALSE, fullrange = FALSE) + theme_classic()

p4

## Warning: position_dodge requires non-overlapping x intervals
```




```
# With mean RA as response and relative abundance as predictor
cor.test(allsumms$mean.prop.dead, allsumms$meanRA, method = "p")

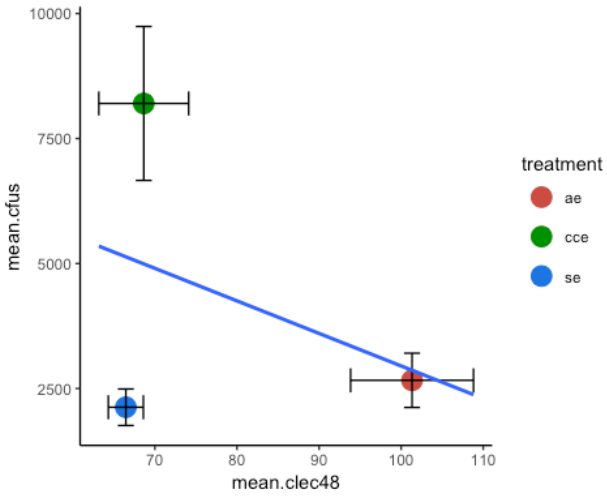
##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.prop.dead and allsumms$meanRA
## t = 1.28, df = 1, p-value = 0.4222
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##       cor
## 0.7880355

summ.cfus.lims <- aes(ymax = allsumms$mean.cfus + allsumms$se.cfus, ymin = allsumms$mea
n.cfus -
    allsumms$se.cfus)
summ.clec48.lims <- aes(xmax = allsumms$mean.clec48 + allsumms$se.clec48, xmin = allsum
ms$mean.clec48 -
    allsumms$se.clec48)

p5 <- ggplot(allsumms, aes(mean.clec48, mean.cfus)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 8) + geom_errorbar(summ.cfus.
lims,
    position = "dodge") + geom_errorbarh(summ.clec48.lims, position = "dodge") +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE) + theme_classic()


p5

## Warning: position_dodge requires non-overlapping x intervals
```



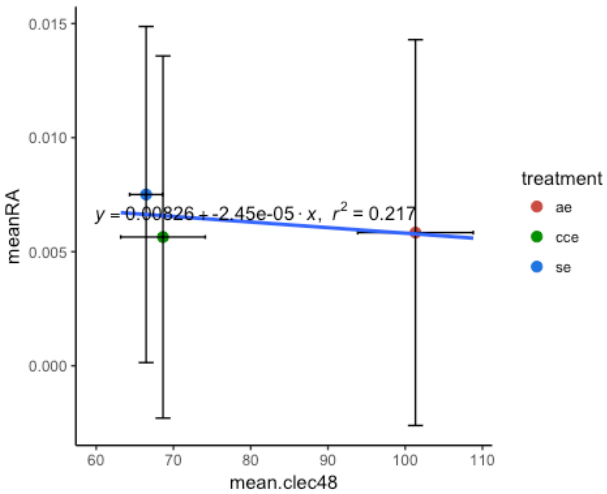

```
# With cfus as esponse and clec-48 transcript abundance as predictor
cor.test(allsumms$mean.cfus, allsumms$mean.clec48, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.cfus and allsumms$mean.clec48
## t = -0.4082, df = 1, p-value = 0.7533
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.3779269

p6 <- ggplot(allsumms, aes(mean.clec48, meanRA)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 4) + geom_errorbar(summ.e.lim
s,
    position = "dodge") + geom_errorbarh(summ.clec48.lims, position = "dodge") +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE) + theme_classic()

p6

## Warning: position_dodge requires non-overlapping x intervals
```



```
# With e faecalis abundnce as response and clec-48 transcript abundance as
# predictor
cor.test(allsumms$meanRA, allsumms$mean.clec48, method = "p")

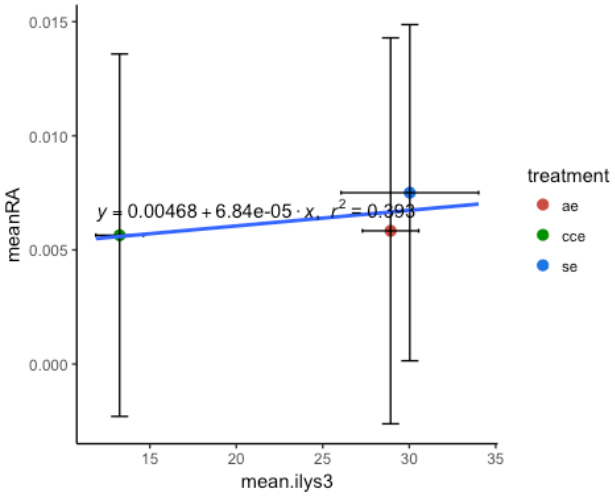
##
##  Pearson's product-moment correlation
##
## data:  allsumms$meanRA and allsumms$mean.clec48
## t = -0.52715, df = 1, p-value = 0.6912
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.4663224

summ.ilys3.lims <- aes(xmax = allsumms$mean.ilys3 + allsumms$se.ilys3, xmin = allsumms$
mean.ilys3 -
    allsumms$se.ilys3)

p7 <- ggplot(allsumms, aes(mean.ilys3, meanRA)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 4) + geom_errorbar(summ.e.lim
s,
    position = "dodge") + geom_errorbarh(summ.ilys3.lims, position = "dodge") +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE) + theme_classic()

p7

## Warning: position_dodge requires non-overlapping x intervals
```



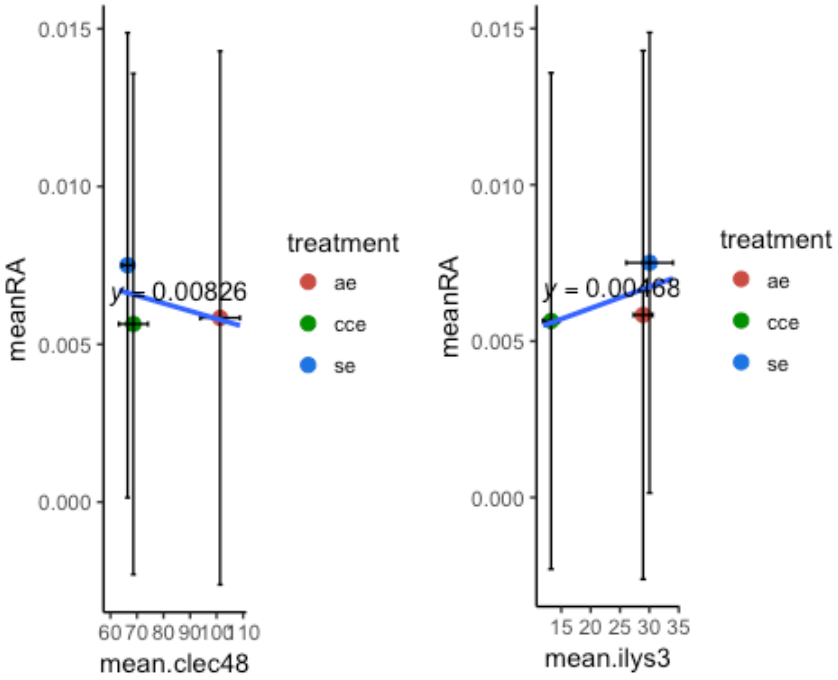

```
# With e faecalis abundnce as response and clec-48 transcript abundance as
# predictor
cor.test(allsumms$meanRA, allsumms$mean.ilys3, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$meanRA and allsumms$mean.ilys3
## t = 0.80432, df = 1, p-value = 0.5688
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##       cor
## 0.6267442

multiplot(p6,p7,cols = 2)
```

```
## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals
```



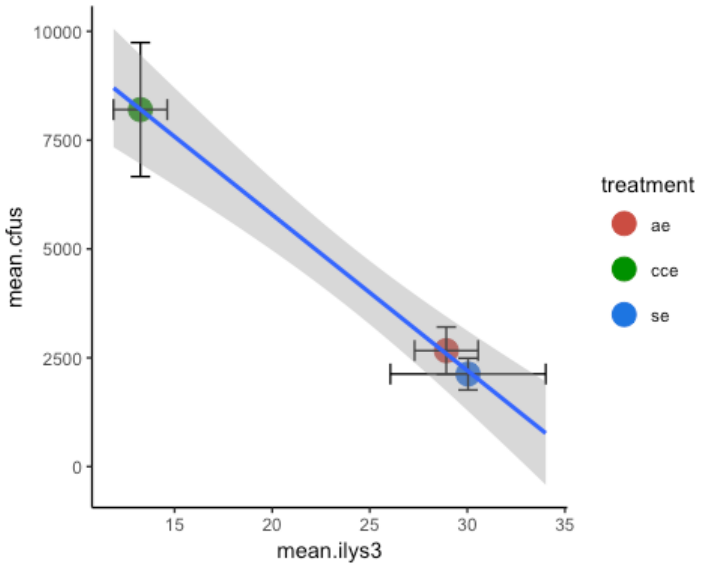
Decrased ilys-3 as a predictor of E. faecalis colonization differences

```
summ.ilys3.lims <- aes(xmax = allsumms$mean.ilys3 + allsumms$se.ilys3, xmin = allsumms$
mean.ilys3 -
    allsumms$se.ilys3)

p8 <- ggplot(allsumms, aes(mean.ilys3, mean.cfus)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 8) + geom_errorbar(summ.cfus.
lims,
    position = "dodge") + geom_errorbarh(summ.ilys3.lims, position = "dodge") +
    geom_smooth(method = lm, se = TRUE, fullrange = TRUE) + theme_classic()


p8

## Warning: position_dodge requires non-overlapping x intervals
```




```
# With cfus as esponse and clec-48 transcript abundance as predictor
cor.test(allsumms$mean.cfus, allsumms$mean.ilys3, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.cfus and allsumms$mean.ilys3
## t = -48.201, df = 1, p-value = 0.01321
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.9997849
```

Check if other genes that were downregulated and associated with immune GO term are associated with increased colonization

```
b0024lims <- aes(xmax = allsumms$mean.B0024.4 + allsumms$se.B0024.4, xmin = allsumms$me
an.B0024.4 -
    allsumms$se.B0024.4)
p9 <- ggplot(allsumms, aes(mean.B0024.4, mean.cfus)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 4) + geom_errorbar(summ.cfus.
lims,
    position = "dodge") + geom_errorbarh(b0024lims, position = "dodge") + geom_smooth(m
ethod = lm,
    se = FALSE, fullrange = TRUE) + ylim(0, 10000) + theme_bw()

Y47H9C.1lims <- aes(xmax = allsumms$mean.Y47H9C.1 + allsumms$se.Y47H9C.1, xmin = allsum
ms$mean.Y47H9C.1 -
    allsumms$se.Y47H9C.1)
p10 <- ggplot(allsumms, aes(mean.Y47H9C.1, mean.cfus)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 4) + geom_errorbar(summ.cfus.
lims,
    position = "dodge") + geom_errorbarh(Y47H9C.1lims, position = "dodge") +
    geom_smooth(method = lm, se = FALSE, fullrange = TRUE) + ylim(0, 10000) +
    theme_bw()

cnc6lims <- aes(xmax = allsumms$mean.cnc6 + allsumms$se.cnc6, xmin = allsumms$mean.cnc6
 -
    allsumms$se.cnc6)
p11 <- ggplot(allsumms, aes(mean.cnc6, mean.cfus)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 4) + geom_errorbar(summ.cfus.
```

```
lims,
    position = "dodge") + geom_errorbarh(cnc6lims, position = "dodge") + geom_smooth(me
thod = lm,
    se = FALSE, fullrange = TRUE) + ylim(0, 10000) + theme_bw()

vhp1lims <- aes(xmax = allsumms$mean.vhp1 + allsumms$se.vhp1, xmin = allsumms$mean.vhp1
 -
    allsumms$se.vhp1)
p12 <- ggplot(allsumms, aes(mean.vhp1, mean.cfus)) + scale_colour_hue(l = 50) +
    geom_point(aes(color = treatment), shape = 20, size = 4) + geom_errorbar(summ.cfus.
lims,
    position = "dodge") + geom_errorbarh(vhp1lims, position = "dodge") + geom_smooth(me
thod = lm,
    se = FALSE, fullrange = TRUE) + ylim(0, 10000) + theme_bw()

multiplot(p9, p10, p11, p12, cols = 2)

## Warning: Removed 2 rows containing missing values (geom_smooth).

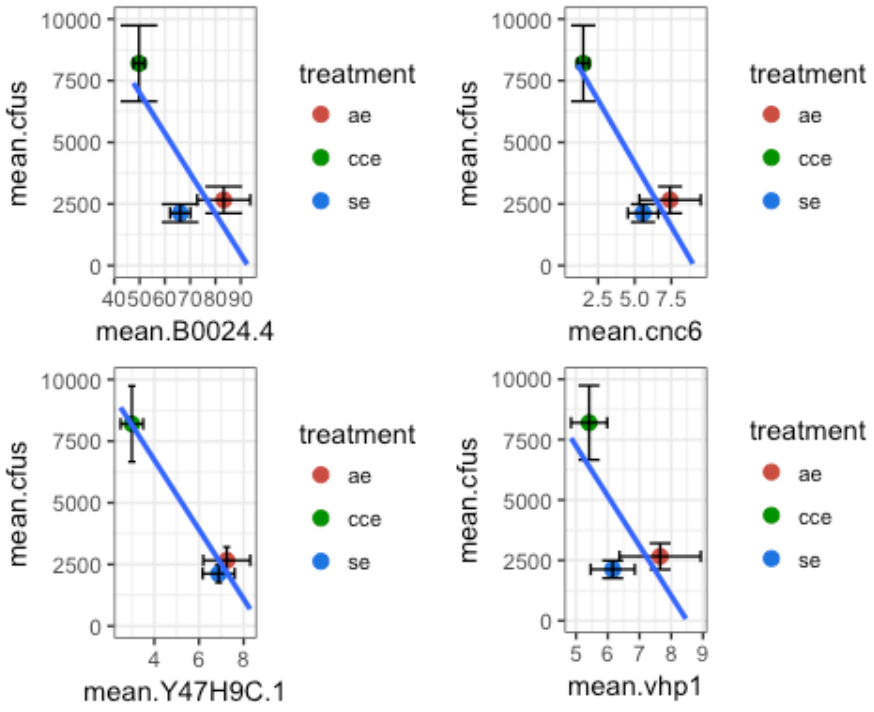
## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: Removed 5 rows containing missing values (geom_smooth).

## Warning: position_dodge requires non-overlapping x intervals

## Warning: Removed 9 rows containing missing values (geom_smooth).
```




```
cor.test(allsumms$mean.cfus, allsumms$mean.B0024.4, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.cfus and allsumms$mean.B0024.4
## t = -1.4008, df = 1, p-value = 0.3947
```

```
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.8138909

cor.test(allsumms$mean.cfus, allsumms$mean.Y47H9C.1, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.cfus and allsumms$mean.Y47H9C.1
## t = -6.3479, df = 1, p-value = 0.09947
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.9878179

cor.test(allsumms$mean.cfus, allsumms$mean.cnc6, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.cfus and allsumms$mean.cnc6
## t = -2.4349, df = 1, p-value = 0.2481
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.9250238

cor.test(allsumms$mean.cfus, allsumms$mean.vhp1, method = "p")

##
##  Pearson's product-moment correlation
##
## data:  allsumms$mean.cfus and allsumms$mean.vhp1
## t = -0.98096, df = 1, p-value = 0.5061
## alternative hypothesis: true correlation is not equal to 0
## sample estimates:
##        cor
## -0.7002768
```

None other are significant predictors.

Worth noting, I make these plots and run correlations to investigate genes for future molecular work and not to make models that are in themselves meaningful. This is because on checking each I clearly have small sample sizes and potential outliers. The same goes for correlations drawn with Enterococcus.

```
devtools::session_info()

## Session info -------------------------------------------------------------

##  setting  value
##  version  R version 3.4.0 (2017-04-21)
##  system   x86_64, darwin15.6.0
##  ui       X11
##  language (EN)
##  collate  en_US.UTF-8
##  tz       America/Los_Angeles
##  date     2017-06-01

## Packages -----------------------------------------------------------------
```

```
##  package              * version  date       source
##  acepack                1.4.1    2016-10-29 CRAN (R 3.4.0)
##  ade4                   1.7-6    2017-03-23 CRAN (R 3.4.0)
##  annotate               1.54.0   2017-04-25 Bioconductor
##  AnnotationDbi          1.38.0   2017-04-25 Bioconductor
##  ape                  * 4.1      2017-02-14 CRAN (R 3.4.0)
##  assertthat             0.2.0    2017-04-11 cran (@0.2.0)
##  backports              1.0.5    2017-01-18 CRAN (R 3.4.0)
##  base                 * 3.4.0    2017-04-21 local
##  base64enc              0.1-3    2015-07-28 CRAN (R 3.4.0)
##  Biobase                2.36.2   2017-05-04 Bioconductor
##  BiocGenerics           0.22.0   2017-04-25 Bioconductor
##  BiocParallel           1.10.1   2017-05-03 Bioconductor
##  biomformat             1.4.0    2017-04-25 Bioconductor
##  Biostrings             2.44.0   2017-04-25 Bioconductor
##  bitops                 1.0-6    2013-08-17 CRAN (R 3.4.0)
##  checkmate              1.8.2    2016-11-02 CRAN (R 3.4.0)
##  cluster                2.0.6    2017-03-10 CRAN (R 3.4.0)
##  codetools              0.2-15   2016-10-05 CRAN (R 3.4.0)
##  colorspace             1.3-2    2016-12-14 CRAN (R 3.4.0)
##  compiler               3.4.0    2017-04-21 local
##  dada2                * 1.4.0    2017-04-25 Bioconductor
##  data.table           * 1.10.4   2017-02-01 CRAN (R 3.4.0)
##  datasets             * 3.4.0    2017-04-21 local
##  DBI                    0.6-1    2017-04-01 CRAN (R 3.4.0)
##  DelayedArray           0.2.2    2017-05-07 Bioconductor
##  DESeq2                 1.16.1   2017-05-06 Bioconductor
##  devtools               1.13.1   2017-05-13 CRAN (R 3.4.0)
##  digest                 0.6.12   2017-01-27 CRAN (R 3.4.0)
##  dplyr                  0.5.0    2016-06-24 cran (@0.5.0)
##  evaluate               0.10     2016-10-11 CRAN (R 3.4.0)
##  foreach                1.4.3    2015-10-13 CRAN (R 3.4.0)
##  foreign                0.8-68   2017-04-24 CRAN (R 3.4.0)
##  formatR                1.5      2017-04-25 CRAN (R 3.4.0)
##  Formula                1.2-1    2015-04-07 CRAN (R 3.4.0)
##  genefilter             1.58.1   2017-05-06 Bioconductor
##  geneplotter            1.54.0   2017-04-25 Bioconductor
##  GenomeInfoDb           1.12.0   2017-04-25 Bioconductor
##  GenomeInfoDbData       0.99.0   2017-05-11 Bioconductor
##  GenomicAlignments      1.12.1   2017-05-12 Bioconductor
##  GenomicRanges          1.28.1   2017-05-03 Bioconductor
##  ggplot2              * 2.2.1    2016-12-30 CRAN (R 3.4.0)
##  graphics             * 3.4.0    2017-04-21 local
##  grDevices            * 3.4.0    2017-04-21 local
##  grid                 * 3.4.0    2017-04-21 local
##  gridExtra              2.2.1    2016-02-29 CRAN (R 3.4.0)
##  gtable                 0.2.0    2016-02-26 CRAN (R 3.4.0)
##  Hmisc                  4.0-3    2017-05-02 CRAN (R 3.4.0)
##  htmlTable              1.9      2017-01-26 CRAN (R 3.4.0)
##  htmltools              0.3.6    2017-04-28 CRAN (R 3.4.0)
##  htmlwidgets            0.8      2016-11-09 CRAN (R 3.4.0)
##  httr                   1.2.1    2016-07-03 CRAN (R 3.4.0)
##  hwriter                1.3.2    2014-09-10 CRAN (R 3.4.0)
##  igraph                 1.0.1    2015-06-26 CRAN (R 3.4.0)
##  IRanges                2.10.1   2017-05-11 Bioconductor
##  iterators              1.0.8    2015-10-13 CRAN (R 3.4.0)
##  jsonlite               1.4      2017-04-08 CRAN (R 3.4.0)
##  knitr                  1.15.1   2016-11-22 CRAN (R 3.4.0)
##  labeling               0.3      2014-08-23 CRAN (R 3.4.0)
##  lattice              * 0.20-35  2017-03-25 CRAN (R 3.4.0)
##  latticeExtra           0.6-28   2016-02-09 CRAN (R 3.4.0)
##  lazyeval               0.2.0    2016-06-12 CRAN (R 3.4.0)
##  limma                * 3.32.2   2017-05-02 Bioconductor
```

```
##  locfit                 1.5-9.1  2013-04-20 CRAN (R 3.4.0)
##  magrittr               1.5      2014-11-22 CRAN (R 3.4.0)
##  MASS                   7.3-47   2017-02-26 CRAN (R 3.4.0)
##  Matrix                 1.2-10   2017-04-28 CRAN (R 3.4.0)
##  matrixStats            0.52.2   2017-04-14 CRAN (R 3.4.0)
##  memoise                1.1.0    2017-04-21 CRAN (R 3.4.0)
##  methods              * 3.4.0    2017-04-21 local
##  mgcv                   1.8-17   2017-02-08 CRAN (R 3.4.0)
##  multtest               2.32.0   2017-04-25 Bioconductor
##  munsell                0.4.3    2016-02-13 CRAN (R 3.4.0)
##  nlme                   3.1-131  2017-02-06 CRAN (R 3.4.0)
##  nnet                   7.3-12   2016-02-02 CRAN (R 3.4.0)
##  parallel               3.4.0    2017-04-21 local
##  permute              * 0.9-4    2016-09-09 CRAN (R 3.4.0)
##  phyloseq             * 1.20.0   2017-04-25 Bioconductor
##  plotly               * 4.6.0    2017-04-25 CRAN (R 3.4.0)
##  plyr                 * 1.8.4    2016-06-08 CRAN (R 3.4.0)
##  purrr                  0.2.2.2  2017-05-11 CRAN (R 3.4.0)
##  R6                     2.2.1    2017-05-10 CRAN (R 3.4.0)
##  RColorBrewer           1.1-2    2014-12-07 CRAN (R 3.4.0)
##  Rcpp                 * 0.12.10  2017-03-19 CRAN (R 3.4.0)
##  RcppParallel           4.3.20   2016-08-16 CRAN (R 3.4.0)
##  RCurl                  1.95-4.8 2016-03-01 CRAN (R 3.4.0)
##  reshape2               1.4.2    2016-10-22 CRAN (R 3.4.0)
##  rhdf5                  2.20.0   2017-04-25 Bioconductor
##  rmarkdown              1.5      2017-04-26 CRAN (R 3.4.0)
##  rpart                  4.1-11   2017-03-13 CRAN (R 3.4.0)
##  rprojroot              1.2      2017-01-16 CRAN (R 3.4.0)
##  Rsamtools              1.28.0   2017-04-25 Bioconductor
##  RSQLite                1.1-2    2017-01-08 CRAN (R 3.4.0)
##  S4Vectors              0.14.1   2017-05-11 Bioconductor
##  scales                 0.4.1    2016-11-09 CRAN (R 3.4.0)
##  ShortRead              1.34.0   2017-04-25 Bioconductor
##  splines                3.4.0    2017-04-21 local
##  stats                * 3.4.0    2017-04-21 local
##  stats4                 3.4.0    2017-04-21 local
##  stringi                1.1.5    2017-04-07 CRAN (R 3.4.0)
##  stringr                1.2.0    2017-02-18 CRAN (R 3.4.0)
##  SummarizedExperiment   1.6.1    2017-05-03 Bioconductor
##  survival               2.41-3   2017-04-04 CRAN (R 3.4.0)
##  tibble                 1.3.0    2017-04-01 CRAN (R 3.4.0)
##  tidyr                  0.6.2    2017-05-04 cran (@0.6.2)
##  tools                  3.4.0    2017-04-21 local
##  utils                * 3.4.0    2017-04-21 local
##  vegan                * 2.4-3    2017-04-07 CRAN (R 3.4.0)
##  viridisLite            0.2.0    2017-03-24 CRAN (R 3.4.0)
##  withr                  1.0.2    2016-06-20 CRAN (R 3.4.0)
##  XML                    3.98-1.7 2017-05-03 CRAN (R 3.4.0)
##  xtable                 1.8-2    2016-02-05 CRAN (R 3.4.0)
##  XVector                0.16.0   2017-04-25 Bioconductor
##  yaml                   2.1.14   2016-11-12 CRAN (R 3.4.0)
##  zlibbioc               1.22.0   2017-04-25 Bioconductor
```