

Why whales are big but not bigger: physiological drivers and ecological limits in the age of ocean giants

Authors: J. A. Goldbogen^{1*}, D. E. Cade¹, D. M. Wisniewska¹, J. Potvin², P. S. Segre¹, M. S.
5 Savoca¹, E. L. Hazen^{3,4}, M. F. Czapanskiy¹, S. R. Kahane-Rapport¹, S. L. DeRuiter⁵, S. Gero⁶, P.
H. Tønnesen⁶, W. T. Gough¹, M. B. Hanson⁷, M. M. Holt⁷, F. H. Jensen⁸, M. Simon⁹, A. K.
Stimpert¹⁰, P. Arranz¹¹, D. W. Johnston¹², D. P. Nowacek¹³, S. E. Parks¹⁴, F. Visser^{15,16,17}, A. S.
Friedlaender⁴, P. L. Tyack¹⁸, P. T. Madsen⁶, N. D. Pyenson^{19,20}

10 **Affiliations:**

¹Hopkins Marine Station, Department of Biology, Stanford University, Pacific Grove, CA, USA

²Department of Physics, Saint Louis University, Missouri, USA

³Environmental Research Division, National Oceanic and Atmospheric Administration,
Southwest Fisheries Science Center, Monterey, CA

15 ⁴Institute of Marine Sciences, University of California, Santa Cruz, Santa Cruz, CA, USA

⁵Mathematics and Statistics Department, Calvin College, Grand Rapids, MI, USA

⁶Zoophysiology, Department of Bioscience, Aarhus University, Aarhus, Denmark

⁷Conservation Biology Division, Northwest Fisheries Science Center, National Marine Fisheries
Service, National Oceanic and Atmospheric Administration, Seattle, WA, USA

20 ⁸Biology Department, Woods Hole Oceanographic Institution, Woods Hole, MA, USA

⁹Greenland Climate Research Centre, Greenland Institute of Natural Resources, Nuuk,
Greenland

¹⁰Moss Landing Marine Laboratories, Moss Landing, CA, USA

¹¹Biodiversity, Marine Ecology and Conservation Group, Department of Animal Biology,
25 University of La Laguna, La Laguna, Spain

¹²Nicholas School of the Environment, Duke University Marine Laboratory, Beaufort, NC, USA

¹³Pratt School of Engineering, Duke University, Durham, NC, USA

¹⁴Department of Biology, Syracuse University, Syracuse, NY, USA

¹⁵Department of Freshwater and Marine Ecology, IBED, University of Amsterdam, the
30 Netherlands

¹⁶Department of Coastal Systems, NIOZ and Utrecht University, the Netherlands

¹⁷Kelp Marine Research, Hoorn, The Netherlands

¹⁸Sea Mammal Research Unit, School of Biology, Scottish Oceans Institute, University of St
Andrews, St Andrews, United Kingdom

35 ¹⁹Department of Paleobiology, National Museum of Natural History, Washington D.C., USA

²⁰Department of Paleontology and Geology, Burke Museum of Natural History and Culture,
Seattle, WA 98105, USA

*Correspondence to: jergold@stanford.edu.

40

45

Abstract:

The largest animals are marine filter feeders, but the underlying mechanism remains unexplained. We measured feeding performance and prey quality to demonstrate how whale gigantism is driven by the interplay of prey abundance and harvesting mechanisms that increase prey capture rates and energy intake. The efficiency of toothed whales that feed on single prey is constrained by the abundance of large prey, whereas filter-feeding baleen whales seasonally exploit vast swarms of small prey at high efficiencies. Given temporally and spatially aggregated prey, filter feeding provides an evolutionary pathway to extremes in body size that are not available to lineages that must feed on one prey at a time. Maximum size in filter feeders is likely constrained by prey availability across space and time. (123/125 words)

One Sentence Summary:

Whale gigantism is driven by specialized structures that increase net energy intake, but maximum body size is limited by the availability of prey. (123/125 characters)

Main Text:

Large body size can improve metabolic and locomotor efficiency. In the oceans, extremely large body size evolved multiple times, especially among edentulous filter feeders that exploit dense patches of small-bodied prey (1, 2). All of these filter feeders had smaller toothed ancestors that targeted much larger, single prey (3, 4). The ocean has hosted the rise and fall of giant tetrapods since the Triassic, but the largest animals of all time persist in today's oceans comprising multiple cetacean lineages (5-8). The evolution of specialized foraging mechanisms that distinguish the two major whale clades—biosonar-guided foraging on individual prey in toothed

whales (Odontoceti) and engulfment filter feeding on prey aggregations in baleen whales
70 (Mysticeti)—likely powered the diversification of crown cetaceans during the Oligocene (~33-23
Ma). The origin of these foraging mechanisms preceded the recent evolution of the largest body
sizes (9, 10), and their diversification across this body size spectrum was likely enhanced by
scale-dependent predator-prey processes (11). It is hypothesized that toothed whales evolved
75 larger body size to enhance diving capacity and exploit deep-sea prey using more powerful
biosonar (12), whereas baleen whales evolved larger sizes for more efficient exploitation of
abundant yet patchily distributed small-bodied prey (13). Cetacean foraging performance is
constrained by diving physiology because they must balance two spatially decoupled resources:
oxygen at the sea surface and higher quality food at depth (14). In both lineages, large body size
confers an ecological benefit that arises from the scaling of fundamental physiological processes;
80 and in some species, anatomical, molecular and biochemical adaptations further enhance diving
capacity (13). As animal size increases, mass-specific oxygen storage is constant yet mass-
specific oxygen usage decreases (13). Therefore, larger air-breathers should have greater diving
capacity, and thus be capable to feed for longer periods at a given depth, leading to higher
feeding rates overall. In theory, this leads to relatively greater dive-specific energy intake with
85 increasing body size, and, with unlimited prey at the scale of foraging grounds and seasons,
larger divers will also exhibit greater energetic efficiencies (i.e., energy intake relative to energy
use) while foraging. We hypothesized that the energetic efficiency of foraging will increase with
body size because larger animals will have greater diving capacities and more opportunities to
feed more frequently per dive. Filter feeding baleen whales will exhibit relatively higher
90 efficiencies compared to single-prey feeding toothed whales because they can exploit greater

biomass at lower trophic levels. This study uses whale-borne tag data to provide a comparative test of these fundamental predictions.

Our direct measures of foraging performance using multi-sensor tags (Fig. 1) show that the largest odontocetes, such as sperm whales (*Physeter macrocephalus*) and beaked whales (Ziphiidae), exhibited high feeding rates during long, deep dives (Fig. 2). By investing time and energy in prolonged dives, these whales accessed deeper habitats that contained less mobile and potentially more abundant prey (15), such as weakly muscularized, ammoniacal squid.

Conversely, rorqual whales performed fewer feeding events per dive despite their large body size, because they invested large amounts of energy to engulf larger volumes of prey-laden water (16). The energetic efficiency (E_E = the energy from captured prey divided by the expended energy including diving costs and post-dive recovery) is determined largely by the number of feeding events per dive (Fig. 2) and the amount of energy obtained during each feeding event (Fig. 3). The amount of energy obtained during each feeding event was calculated from prey type and prey size distributions historically found in the stomachs of odontocetes (except for killer whales: we used identified prey remains from visually confirmed prey capture events), as well as the acoustically measured biomass, density and distribution of krill at rorqual foraging hotspots (17). Our results show that although larger odontocetes appear to feed on larger prey relative to smaller toothed whale prey, they were not disproportionately larger (Fig. 3; Table S11) and toothed whales did feed more frequently on this smaller prey type; thus, the energy obtained from prey in a dive did not outweigh the increased costs associated with larger body size and deeper dives (Fig. S2), thereby causing a decrease in E_E with increasing body size (Fig. 4). In contrast, the measured distribution and density of krill biomass suggests that larger rorquals are

not prey-limited at the scale of individual dives. Because larger rorquals have relatively larger
115 engulfment capacities (16), rorquals exhibited much more rapid increases in energy captured
from prey with increasing body size (Fig. 3). If they can detect and exploit the densest parts of an
individual krill patch, as evidenced by their ability to maneuver more and increase feeding rates
per dive when krill density is higher (14), then E_E should increase with body size (Fig. 4). We
note that these results were robust to assumptions about trait similarity from shared ancestry as
120 well as the scaling of metabolic rate (MR), which we simulated over a wide range as
($MR \propto Mc^{0.45:0.75}$).

The divergence in energetic scaling between rorquals and odontocetes that results from available
prey has major implications for understanding the ecology and evolution of gigantism in marine
125 ecosystems. For toothed whales, increasing body size leads to hyper-allometric investment in
biosonar structures that increase prey detection range (12). The largest living toothed whales
today, sperm whales and beaked whales, independently evolved large body size to push their
physiological limits for dive duration to spend more time feeding in the deep sea. The
mesopelagic and bathypelagic realms are not only among the largest ecosystems on the planet,
130 they also provide less competitive niches with fewer endothermic predators, providing
opportunities to capture high-value prey (18). Although sperm whales foraging on giant squids
(Architeuthidae) persists as an iconic motif, giant squid beaks are rare in sperm whale stomachs
at a global scale (19). However, sperm whale biosonar is, owing to a hypertrophied nasal
complex, more powerful than beaked whale biosonar by approximately two orders of magnitude
135 (12), which allows sperm whales to scan larger volumes of water and in some regions to find and
chase very large prey. Sperm whales have higher attack speeds and reduced feeding rates per

dive when foraging on giant squid (20), which contrasts with how sperm whales feed with slower speeds and higher rates on smaller squid in other regions (21). This discrepancy suggests that larger prey will incur greater foraging costs which partially offsets the increased energetic gain.

140 Smaller prey are usually more abundant than larger prey (22), so efforts to optimize foraging efficiency require the ability to detect the distribution of prey size, which favors the evolution of powerful sonar. Both beaked whales and many sperm whales in our study may have adopted a less risky strategy by targeting more reliable patches of cephalopods often at depths greater than 1,000 m, thereby yielding up to 50 feeding events per dive (Fig. 2). Nevertheless, the ability of
145 sperm whales to forage on the largest squid, when available, highlights an advantage of their large size compared to beaked whales, which feed on smaller prey. Regardless of whether odontocetes target a few large prey or many small prey in individual dives, the energy gained from these unique deep-sea resources is ultimately constrained by the total amount of prey biomass that can be captured during a breath-hold dive. Therefore, prey availability is a key
150 ecological factor that constraints body size and population density in these lineages.

By contrast, gigantism in mysticetes is advantageous because they exhibit positive allometry in filter-feeding adaptations that enable bulk consumption of dense prey patches (16). For the largest rorquals, each lunge captured a patch of krill with an integrated biomass and energetic
155 content that exceeded, on average, those of the largest toothed whale prey by at least an order of magnitude (Fig. 3). This ability to process large volumes of prey-laden water, calculated as 100-160% of the whale's own body in the largest rorquals, underlies the high energetic efficiency of foraging even when accounting for differences in body size (Fig. S1). During lunge feeding, water and prey are engulfed in a matter of seconds and at speeds several times higher than steady

160 swimming (16). However, whales in a separate mysticete clade (Balaenidae), represented by
bowhead whales (*Balaena mysticetus*) and right whales (*Eubalaena* spp.), do not feed in discrete
events but rather continuously ram prey-laden water through their baleen for up to several
minutes at a time (23). The speed-dependent drag associated with continuous ram filtration
necessitates slow swimming speeds to minimize energy expenditure (23). This strategy may be
165 optimized for foraging on smaller copepods that form less dense patches, thereby resulting in
lower energetic efficiencies relative to similarly sized rorquals (Fig. 4). The high-speed dynamics
of rorqual lunge feeding also generate high drag (16), but the rapid engulfment of dense krill
patches yields higher efficiencies. Both continuous ram filter-feeding and lunge-feeding
mysticetes appeared to have independently evolved gigantism (>12 m body length) during an era
170 of intensified wind-driven upwelling and glacial cycles, processes that characterize productive
whale foraging hotspots in the modern oceans (9). Coastal upwelling intensity increases the
number and density of aggregations of the relatively small-bodied forage species (24) that make
filter feeding energetically efficient (14). Our analyses point to filter feeding as a mechanism that
explains the evolutionary pathway to gigantism because it enabled the high efficiency
175 exploitation of large, dense patches of prey.

The largest comparable vertebrates, sauropod dinosaurs, reached their maximum size on land
about mid-way through their 140-million-year history, and their evolutionary patterns show no
real limits to extreme size (25). If sauropod size was not limited by physical factors such as
180 gravity, hemodynamics and bone mechanics (26), then it may have been ultimately constrained
by energetics and food availability (27), rather than an ability to access available food. In the
marine environment, the combination of filter feeding and greater abundance of food likely

facilitated the evolution of not only gigantic filter feeding whales, but also several independent lineages of large filter feeding elasmobranchs (3, 6). Both filter feeding sharks and mesothermic single-prey feeding sharks exhibit greater body size compared to single-prey feeding ectothermic sharks (3), suggesting parallel evolutionary trajectories with cetaceans in terms of gigantism and morphological adaptations that increase foraging capacity and net energy intake (4). The largest filter feeding sharks are larger than mesothermic raptorial feeding sharks, which may reflect either a lack of large prey as a limiting factor in today's oceans or an additional temperature-dependent metabolic constraint. Similarly, the larger size of baleen whales compared to filter feeding sharks suggests an overall advantage for animals that exhibit both endothermy and filter feeding adaptations - particularly in cold, productive habitats. The combination of high metabolic rates and the ability to short circuit the food web with filter feeding adaptations may have enabled high-efficiency exploitation of low trophic levels (28), thereby facilitating the evolution of large body size in multiple lineages.

We have shown that cetacean gigantism is driven by the hyper allometry of structures that increase prey capture rates and energy intake in clades with divergent feeding mechanisms despite the potential constraints to size. However, to maintain a high energetic efficiency at larger sizes, cetaceans must exploit either large individual prey or dense patches of small prey. Although the lack of large prey and the increasing costs of capturing such prey limits energetic efficiency of the largest toothed whales, our analyses suggest that large rorquals are not limited by the size and density of krill patches at the productive apex of their foraging seasons. How long these dense krill patches are available during the summer feeding season at higher latitudes, or throughout the rest of the year (29), may ultimately determine the amount of lipid reserves

that can be used to fuel ocean basin-scale migrations as well as their reproductive output at lower latitudes (30, 31). The size of the largest animals does not seem to be limited by physiology (5), but rather is limited by prey availability and the rate at which that prey can be exploited using the foraging mechanisms they have evolved.

210

Acknowledgments: We thank Chris Taylor for Echoview processing and active acoustic data collection, Alex Boersma for providing illustrations of cetacean species and their prey and Per Henriksen for photo of squid beaks and sperm whale tooth. We also thank William Gearty for assistance with comparative phylogenetic analyses.

215

Funding: This research was funded in part by grants from the National Science Foundation (IOS-1656676, IOS-1656656; OPP-1644209 and 07-39483), the Office of Naval Research (N000141612477), and a Terman Fellowship from Stanford University. All procedures in USA were conducted under approval of the National Marine Fisheries Service (Permits 781-1824, 16163, 14809, 16111, 19116, 15271, 20430), Canada DFO SARA/MML 2010-01/SARA-106B, National Marine Sanctuaries (MULTI-2017-007), Antarctic Conservation Act (2009-014, 2015-011) and institutional IACUC committee protocols. Fieldwork, data collection and data processing for *M. densirostris* were funded by the Office of Naval Research grants N00014-07-10988, N00014-07-11023, N00014-08-10990, N00014-18-1-2062, and 00014-15-1-2553, and the U.S. Strategic Environmental Research and Development Program Grant SI-1539. PLT gratefully acknowledges funding from funding the MASTS pooling initiative (The Marine Alliance for Science and Technology for Scotland). MASTS is funded by the Scottish Funding Council (HR09011) and contributing institutions. Fieldwork, data collection and data processing

225

for *G. melas* and data collection at the Azores were funded by the Office of Naval Research
230 (ONR grants N00014-12-1-0410, N000141210417, N00014-15-1-2341, N00014-17-1-2715), the
Danish Council for Independent Research (award number 0602-02271B) and the Dutch Research
Council (award number 016.Veni.181.086). SRKW field work: We thank NOAA Ocean
Acoustics Program for providing funding and Candice Emmons, Deborah Giles and Jeffery
Hogan for assistance in the field. For *P. macrocephalus* fieldwork off Dominica was supported
235 through a FNU fellowship from the Danish Council for Independent Research supplemented by a
Sapere Aude Research Talent Award, a Carlsberg Foundation expedition grant, a grant from
Focused on Nature, and a CRE Grant from the National Geographic Society to SG; a FNU large
frame grant, as well as a Villum Foundation Grant (PTM) with supplementary grants from Dansk
Akustisk Selskab (PHT), Oticon Foundation (PHT), and Dansk Tennis Foundation (PHT). The
240 Greenland data collection and analysis was funded by Grants from the Oticon Foundation and
the Carlsberg Foundation to MS. Tagging work on *P. phocoena* was funded in part by the
German Federal Agency for Nature Conservation (BfN) under contract Z1.2-530/2010/14 and
the BfN-Cluster 7 “Effects of underwater noise on marine vertebrates”.

245 **Author contributions:** Overall idea, concept and approach developed by JAG. Bioenergetic
models developed by JP and implemented by JAG. Integration of data analysis and interpretation
directed by JAG. Data analysis was conducted and implemented by JAG, DEC, DMW, JP, PSS,
with contributions for statistics by DMW, SLD, MSS, ELH. Manuscript written by JAG and
NDP with contributions by DEC, DMW, JP, PLT, PTM, and FHJ. All authors read, edited, and
250 discussed the manuscript. All authors participated in data collection.

Competing interests: Authors declare no competing interests

Data and materials availability: The data analyzed in this study will be available at the

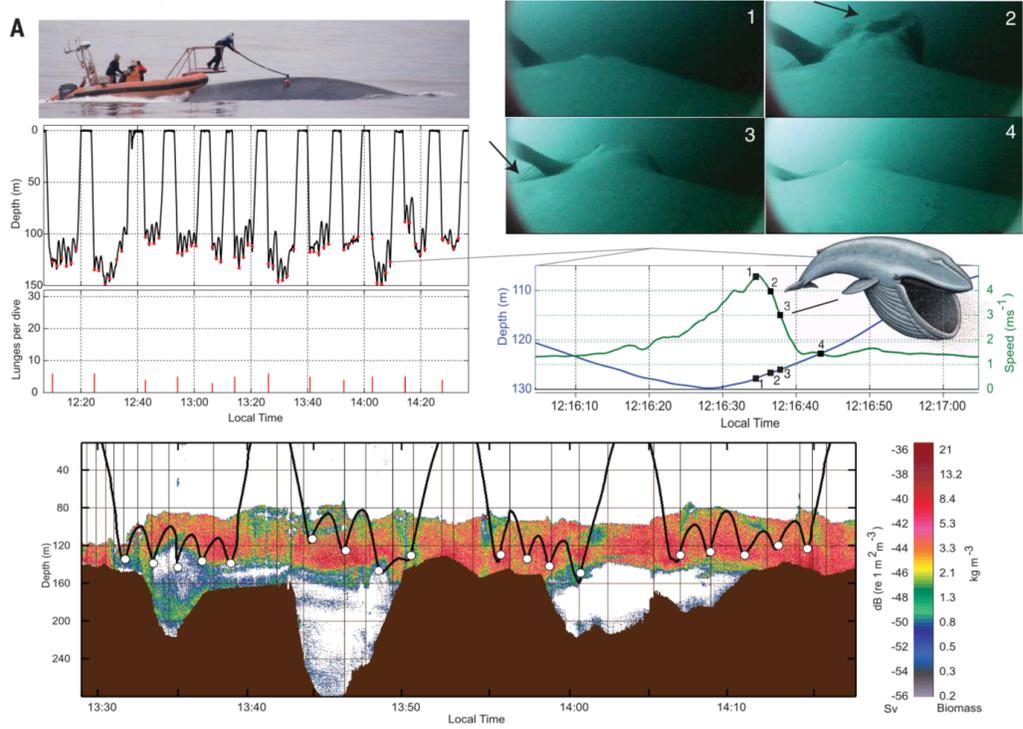
255 Stanford Data Repository (sdr.stanford.edu) immediately upon publication at

<https://purl.stanford.edu/zk778rt5347>

260

265

270



275

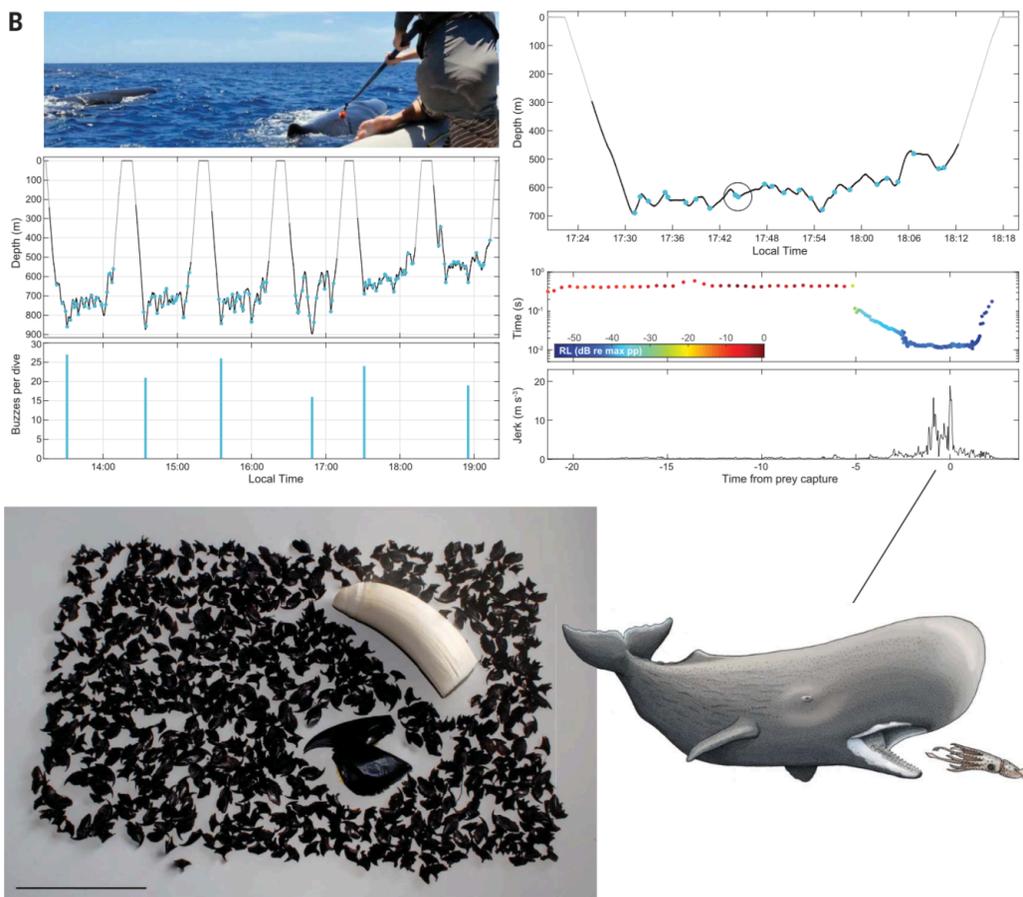


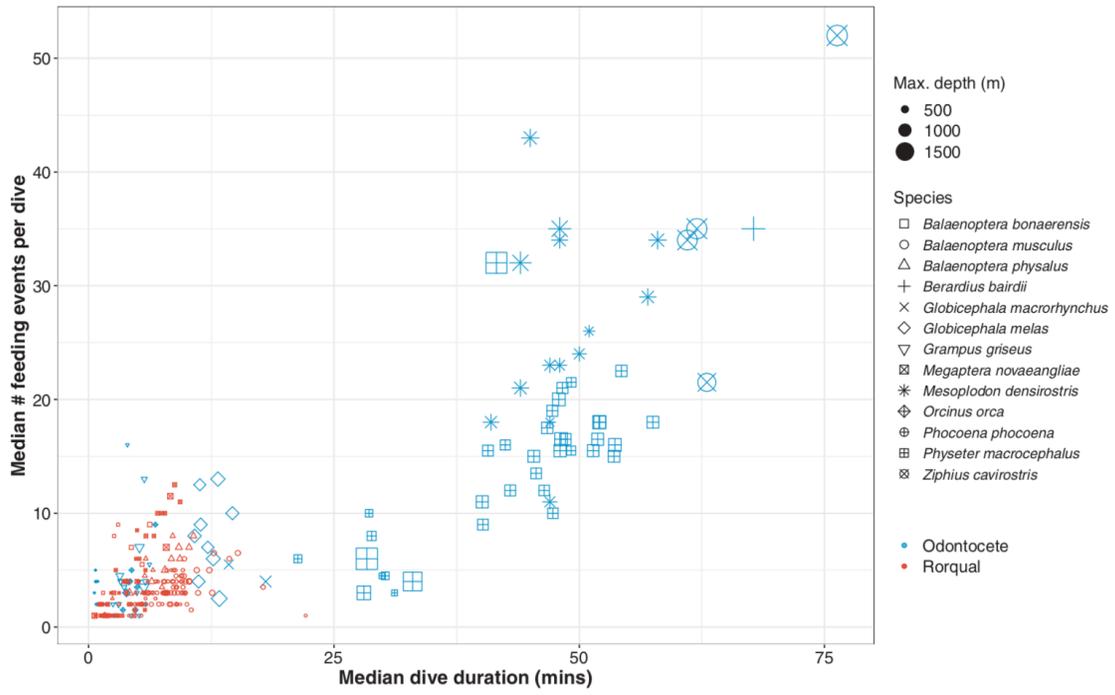
Figure 1. Whale tag data quantifies foraging performance. (A) Blue whale suction-cup tagging using a rigid-hulled inflatable boat and a carbon fiber pole (upper left). Tag data from a blue whale showing 12 consecutive foraging dives and the number of lunge-feeding events per dive (left). Inset right shows the kinematic signatures used to detect lunge feeding events (with an increase in speed and upward movement prior to lunging) and simultaneous video frames that directly confirm engulfment (1-4: with arrows denoting 1=prior to mouth opening, 2=maximum gape, 3= maximum extension of the ventral groove blubber, 4=after mouth closure during the filter phase). Example of time synchronized dive profile and the estimated biomass as a function of depth (see supplemental methods), grid lines are 147 m by 40 m. Prey mapping data was used to estimate the distribution of krill densities targeted by tagged whales. (B) Sperm whale suction-cup tagging (upper left) and 6 foraging dives with feeding events (thicker lines denote echolocation activity). Middle right panels show the acoustic inter-click interval (ICI) and kinematic signatures (jerk, or rate of acceleration) used to infer feeding events at depth. Examples of cephalopod beaks (single large beak, *Mesonychoteuthis hamiltoni*; many small beaks, *Gonatus fabricii*) found in the stomachs of sperm whales (lower left panel) that were used to estimate the size distributions of captured prey (sperm whale tooth and 10 cm line also shown for scale, photo by Per Henriksen). Illustrations by Alex Boersma.

280

285

290

295



300

305

310

Figure 2. Number of feeding events per foraging dive. Beaked whales (*Ziphiidae*) and some sperm whales (*Physeter macrocephalus*) exhibit high feeding rates during long and deep dives, whereas rorquals and delphinids feed less frequently during shorter, shallower dives. We note that balaenids were excluded from this analysis because they are continuous ram filter feeders and do not exhibit discrete feeding events like rorquals and odontocetes.

315

320

325

330

335

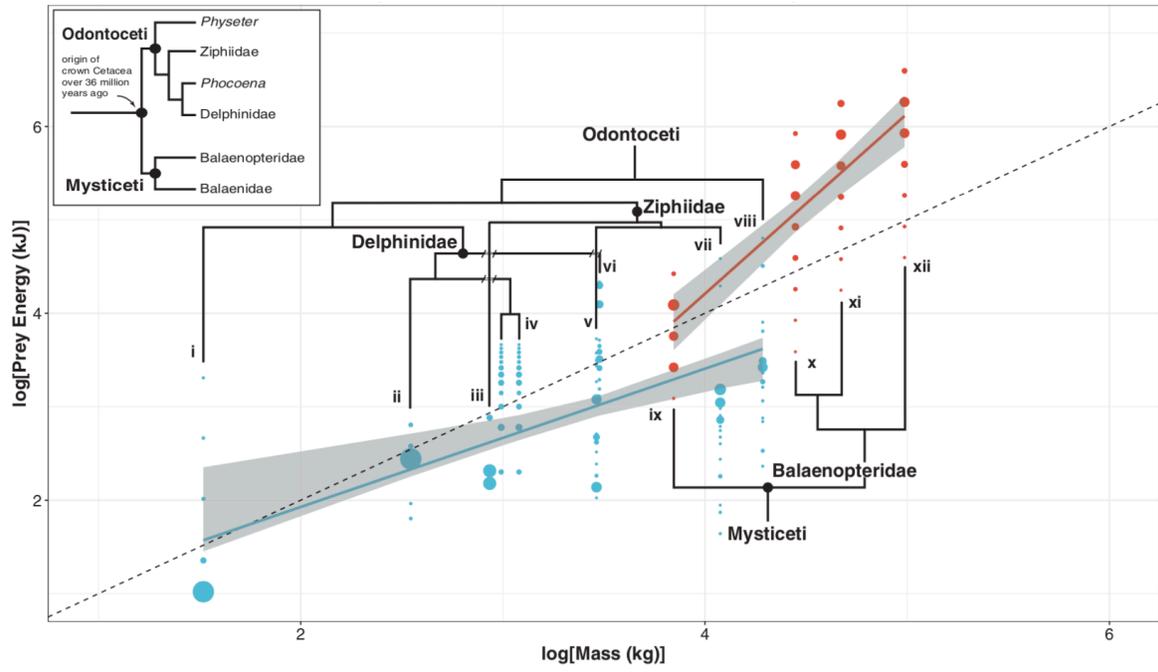
340

345

350

355

360



365

370

375

Figure 3. Scaling of prey energy captured during by toothed whales and rorqual whales during each feeding event. Estimates for prey energy (prey mass x prey energy density)

obtained from each feeding event. For rorquals, the values indicate the integrated energy of all

380 krill captured for each engulfment event. Symbol size indicates the relative frequency of

occurrence based on stomach content data and prey mapping data for odontocetes and

mysticetes, respectively, and this data (the vertical spread of the data reflects the distribution of prey data for each species.) was used to weight the regression (fitted to species-specific means).

The dashed line denotes isometry, indicating that larger toothed whales capture disproportionately

385 less energy from prey ($y = 2.81x^{0.74}$), whereas larger rorquals capture disproportionately larger

prey energy, with increasing body size ($y = 0.000309x^{1.93}$). GLS regressions are shown with 95%

CI (gray bands; see also Table S11). Phylogenetic tree with arbitrary branch lengths showing

evolutionary relationships (32) among species (i, harbor porpoise, *P. phocoena*; ii, Risso's

dolphin, *G. griseus*; iii, Blainville's beaked whale, *M. densirostris*; iv, pilot whales,

390 *Globicephala* sp.; v, Cuvier's beaked whale, *Z. cavirostris*; vi, killer whale, *O. orca*; vii, Baird's

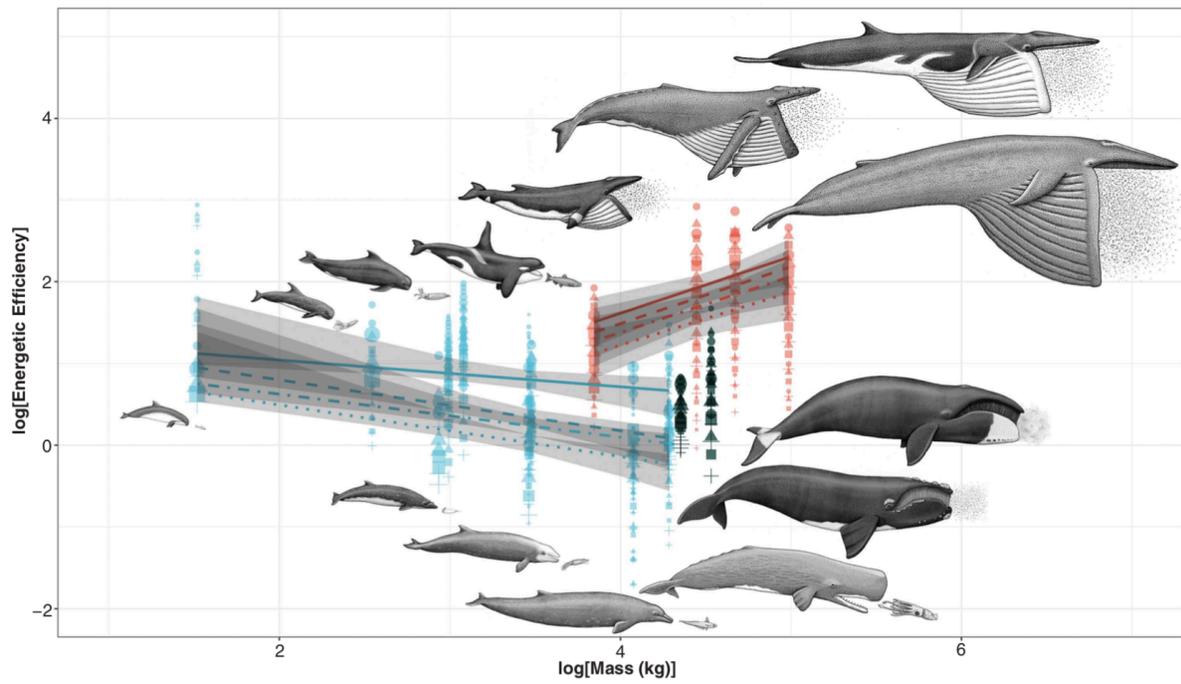
beaked whale, *B. bairdii*; viii, sperm whale, *P. macrocephalus*; ix, Antarctic minke whale, *B.*

bonaerensis; x, humpback whale, *M. novaeangliae*; xi, fin whale, *B. physalus*; xii, blue whale, *B.*

musculus). We note that balaenids were excluded from this analysis because they are continuous

ram filter feeders and do not exhibit discrete feeding events like rorquals and odontocetes.

395



400

405

410

Figure 4. Scaling of energetic efficiency for foraging dives and corresponding surface

415 **intervals.** The energetic efficiency (E_E = the energy from captured prey divided by the expended energy including diving costs and post-dive recovery) of foraging decreases in toothed whales (blue) but increases in rorqual whales lunge filter-feeding on krill (red). Bowhead whales and right whales continuous ram filter-feeding on copepods (green) exhibit lower energetic efficiencies compared to rorqual whales of similar size. These scaling relationships (Table S11)

420 are robust to assumptions about metabolic rate (plus symbols and dotted line, $MR \propto Mc^{0.75}$; squares and dot-dash line, $MR \propto Mc^{0.68}$; triangles and dashed line, $MR \propto Mc^{0.61}$; circles and solid line, $MR \propto Mc^{0.45}$) that modulate the rate of energy expenditure of foraging. Regressions are shown with 95% CI (gray bands). Vertical spread of the data corresponds to prey quality distribution data (as in Fig. 3), with larger icons denoting greater proportions of observed values.

425 The vertical spread of the data reflects the distribution of prey data for each species. Note that log energetic efficiencies less than zero suggest that whales will be unable to survive on that prey type and quality alone. Illustrations by Alex Boersma.

430

List of Supplementary Materials:

Materials and Methods

435 Figures S1, S2

Tables S1-S11

References (*1-185*)

440

445

450

455

460

Supplementary Materials for

465 Why whales are big but not bigger: physiological drivers and ecological limits in
the age of ocean giants

Authors: J.A. Goldbogen^{1*}, D. E. Cade¹, D. M. Wisniewska¹, J. Potvin², P.S. Segre¹, M. S.
Savoca¹, E. L. Hazen^{3,4}, M. F. Czapanskiy¹, S. R. Kahane-Rapport¹, S. L. DeRuiter⁵, S. Gero⁶, P.
470 Tønnesen⁶, W. T. Gough¹, M. B. Hanson⁷, M. Holt⁷, F. H. Jensen⁸, M. Simon⁹, A. K. Stimpert¹⁰,
P. Arranz¹¹, D. W. Johnston¹², D. P. Nowacek¹³, S. E. Parks¹⁴, F. Visser^{15,16,17}, A. S.
Friedlaender⁴, P. L. Tyack¹⁸, P. T. Madsen⁶, N. D. Pyenson^{19,20}

Affiliations:

475 ¹Hopkins Marine Station, Department of Biology, Stanford University, Pacific Grove, CA, USA

²Department of Physics, Saint Louis University, Missouri, USA

³Environmental Research Division, National Oceanic and Atmospheric Administration,

Southwest Fisheries Science Center, Monterey, CA

⁴Institute of Marine Sciences, University of California, Santa Cruz, Santa Cruz, CA, USA

480 ⁵Mathematics and Statistics Department, Calvin College, Grand Rapids, MI, USA

⁶Zoophysiology, Department of Bioscience, Aarhus University, Aarhus, Denmark

⁷Conservation Biology Division, Northwest Fisheries Science Center, National Marine Fisheries
Service, National Oceanic and Atmospheric Administration, Seattle, WA, USA

⁸Biology Department, Woods Hole Oceanographic Institution, Woods Hole, MA, USA

485 ⁹Greenland Climate Research Centre, Greenland Institute of Natural Resources, Nuuk,
Greenland

¹⁰Moss Landing Marine Laboratories, Moss Landing, CA, USA

¹¹Biodiversity, Marine Ecology and Conservation Group, Department of Animal Biology,
University of La Laguna, La Laguna, Spain

490 ¹²Nicholas School of the Environment, Duke University Marine Laboratory, Beaufort, NC, USA

¹³Pratt School of Engineering, Duke University, Durham, NC, USA

¹⁴Department of Biology, Syracuse University, Syracuse, NY, USA

¹⁵Department of Freshwater and Marine Ecology, IBED, University of Amsterdam, the
Netherlands

495 ¹⁶Department of Coastal Systems, NIOZ and Utrecht University, the Netherlands

¹⁷Kelp Marine Research, Hoorn, The Netherlands

¹⁸Sea Mammal Research Unit, School of Biology, Scottish Oceans Institute, University of St
Andrews, St Andrews, United Kingdom

¹⁹Department of Paleobiology, National Museum of Natural History, Washington D.C., USA

500 ²⁰Department of Paleontology and Geology, Burke Museum of Natural History and Culture,
Seattle, WA 98105, USA

*Correspondence to: jergold@stanford.edu.

505

This PDF file includes:

Materials and Methods

510 Supplementary Text

Figure S1-S2

Tables S1-S11

Table of Contents for Materials and Methods:

- 515
1. Toothed whale prey data supplement
 2. Rorqual whale prey data supplement
 3. Balaenidae prey data supplement
 4. Tag data supplement: Using whale-borne tags to quantify foraging rates.
 5. Bio-energetic modeling of foraging costs, gains, and efficiency

520

 6. Statistical Analyses
 7. R-markdown details for comparative phylogenetic analyses

List of Supplemental Figures and Tables:

525 Figure S1. Generalized additive mixed model partial plots for the effect of (baleen and toothed) whale body size and prey intake (for either feeding mode) on the energetic efficiency of a foraging bout.

Figure S2. Scaling of energetic gains and costs for foraging dives and corresponding surface intervals in toothed whales, rorqual whales, and balaenid whales.

Table S1. Estimated lengths of fish targeted by tagged *P. phocoena*

530 Table S2. Calculated weights, energy density and total energy content of fish (*Clupea harengus*)
targeted by tagged *P. phocoena* for energetics modeling.

Table S3. Prey size distribution and calorific value for *G. griseus* energetics modeling.

Table S4. Prey size distribution and calorific value for *M. densirostris* energetics modeling.

Table S5. Prey size distribution and calorific value for *Globicephala* energetics modeling.

535 Table S6. Prey size distribution and calorific value for *Z. cavirostris* energetics modeling.

Table S7. Fish (*Oncorhynchus* sp.) weight, energy density and total energy content used for *O. orca* energetics modeling.

Table S8. Prey size distribution and calorific values used for *B. bairdii* energetics modeling.

Table S9. Prey size distribution and calorific values used for *P. macrocephalus* energetics
540 modeling.

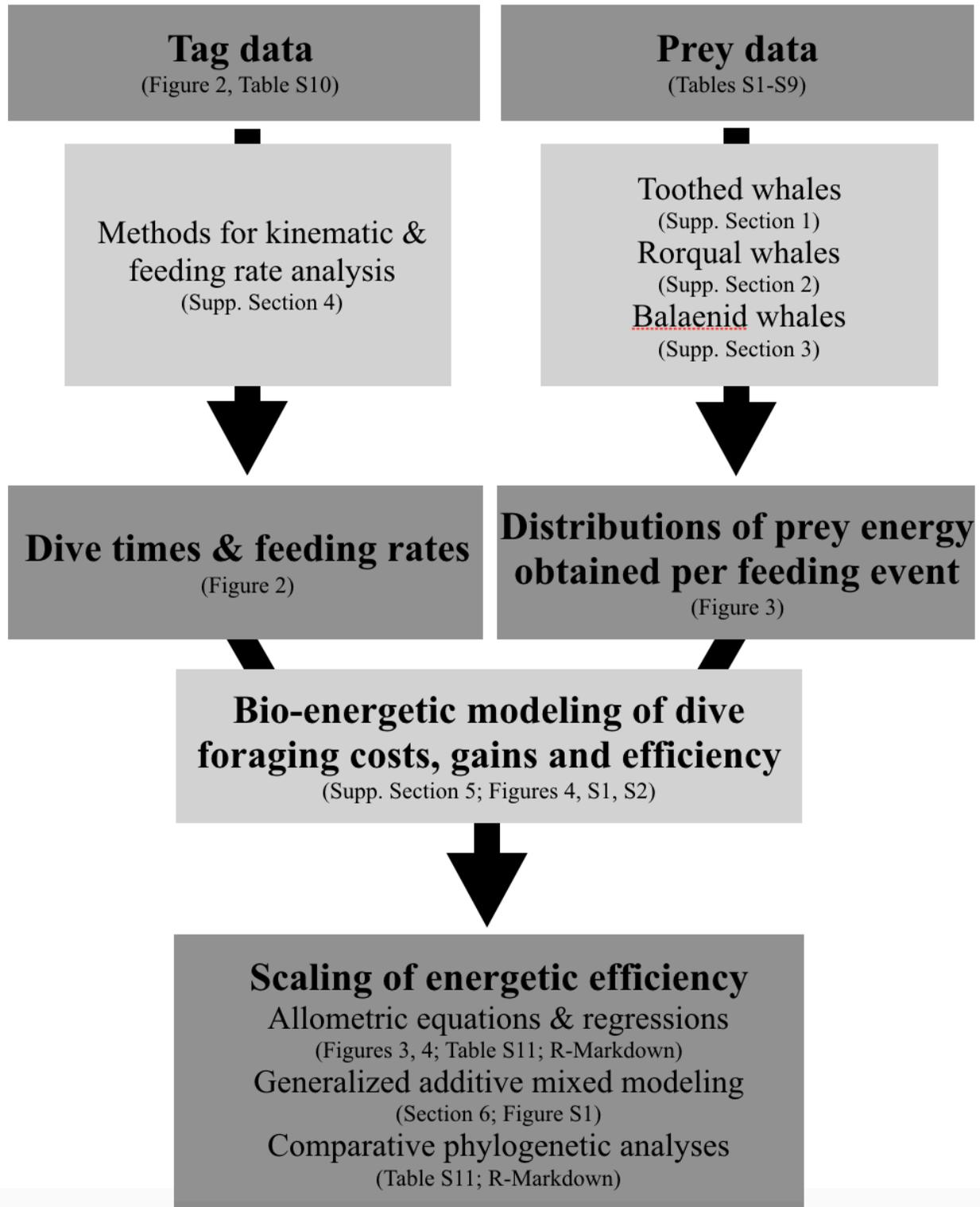
Table S10. Foraging data set from suction-cup tagged cetaceans.

Table S11. Results of phylogenetic (pGLS) and ordinary (OLS) least squares analysis of
cetacean foraging capacity.

545

550

Overall Materials and Methods Outline:



1. Toothed whale prey data supplement

560 1.1. Harbor porpoise (*Phocoena phocoena*): Based on the same tag data used in this study to obtain foraging performance and feeding rates, Wisniewska et al. (33) used acoustic data (i.e. echograms recorded by the whale-attached tags) to detect the swimming strokes of fish attempting to escape from the tagged whale. Assuming prey were escaping at their maximum capacities, Wisniewska et al. (33) estimated a range of fish size 3-10 cm in total length given the relationship between stroke frequency and size in fish (34). The estimated distribution of fish size using this method is shown in Table S1. The primary prey was assumed to be Atlantic
565 herring (*Clupea harengus*) based on stomach content analyses of by-caught whales (35), and we used established length-weight relationships (36) to calculate the weight of captured fish. We used the regressions that relate energy density to fish body mass for herring to estimate the total energy content of targeted fish (Table S2) across scale (37).

570 1.2. Risso's dolphin (*Grampus griseus*): Stomach content data that closely corresponds to the location of tagged whales was not available. We used the nearest location with available data (38). *Histioteuthis reversa* and *Todarodes sagittatus* were the dominant prey species with mean dorsal mantle lengths 79 mm and 177 mm, and weights of 119 g and 210 g, respectively. These size classes are similar to the dominant prey species (*Enoploteuthis*; *Megalocranchia*) found in
575 *G. griseus* in other regions, such as Hawaii (39). We used the regressions that relate beak morphology, dorsal mantle length, and weight to estimate the size distributions (Table S3) of cephalopod prey (40, 41).

580 1.3. Blainville's beaked whale (*Mesoplodon densirostris*): There is very little data on stomach contents and diet in *M. densirostris* (42), but tag data has demonstrated that they do not chase prey over long distances and approach to capture prey at slow speeds within the oxygen minimum zone (43, 44). Santos et al. (2007) reported prey in the stomach of a stranded *M. densirostris*, containing unidentifiable fish and cephalopods (Octopoteuthidae, Histioteuthidae, Cranchiidae). The cranchiid and histioteuthid squid were the dominant cephalopod prey, which with respect to the former type was similar to that found in another *M. densirostris* in more temperate waters (45). Santos et al. (2007) only reported size data for 7 prey items, which again underlines the overall dearth of stomach contents data for this species. Therefore, our prey size distribution only represents a first approximation (Table S4).

590 1.4. Long-finned pilot whale (*Globicephala melas*) and short-finned pilot whale (*Globicephala macrorhynchus*): Although loliginid squid (*Loligo pealei*) are the dominant prey of *G. melas* in the western north Atlantic (46-48), ommastrephid cephalopods (*Todarodes sagittatus*) appear to be dominant prey in the eastern north Atlantic (49). In contrast to short-finned pilot whales (*G. macrorhynchus*) that likely target larger, agile squid (*Todarodes sagittatus*) with sustained sprints at depth (50), tagged *G. melas* exhibited much lower speeds that were closer to cruising speeds during foraging. Due to a lack of prey data for *Globicephala* sp. in the North Atlantic (51), and the lack of provided complete data sets (49), we digitized the data set shown in Figure 2 of Piatowski et al. (52) from trawls, hand-jigging, and commercial bycatch operations in the central eastern Atlantic Ocean (Figure S5) and used this distribution of squid size for energetic modeling. We note that the mean values (DML=250±60 mm) from Piatowski et al. (1998) are

595
600

comparable to the mean squid size reported (DML=200 mm) by Desportes and Mouritsen (1988).

605 1.5. Cuvier's beaked whale (*Ziphius cavirostris*): Much more diet data exists for *Z. cavirostris* compared to many other beaked whale species (53-55). To estimate the range of foraging efficiencies across the full distribution of prey types and sizes in *Z. cavirostris*, we combined the prey data from Santos et al. 2001 (as reported in Table 3 of MacLeod et al. 2003) and West et al. 2017 (Table S6).

610 1.6. Southern resident killer whale (*Orcinus orca*): Based on collected prey or fecal samples during focal follows, Southern resident killer whales (SRKW) of the coastal areas of the eastern North Pacific Ocean have a very specific prey preferences (56-58): chinook salmon (*Oncorhynchus tshawytscha*) and other salmonids including coho salmon (*O. kisutch*). In the late summer when tag data collection occurred (59), they roughly eat 50% *O. tshawytscha* and 50% 615 *O. kisutch* (60, 61) in the size classes summarized in Table S7 (also from NWFSC unpublished data). Whales tend to forage together and also share captured prey (57). Ford and Ellis (2006) reported a range of 2 to 6 whales that forage in groups, so we assumed that an average of 4 whales used the same amount of energy to forage and also shared the energetic value of captured prey among the same 4 whales. To estimate the range of possible foraging efficiencies on *O.* 620 *tshawytscha*, we used the prey distribution shown in Table 12 of Ford et al. 2010 and the age-class-specific size and energy content data from Table 10 in Ford et al. 2010 and Table 8 from Ford & Ellis (2006). To estimate the range of possible foraging efficiencies on *O. kisutch*, we used the mean values and standard deviation reported by O'Neill et al. (2014; see their Table 2

for weight and regression for intermediate lipid species in their figure 3) to derive a distribution
625 of likely prey sizes and energy shown below in Table S7.

1.7. Baird's beaked whale (*Berardius bairdii*): Like most beaked whale species, the prey
preferences of *B. bairdii* is poorly understood and only reported for a few select regions and
individuals (41). Whales caught off the coast of Japan generally exhibited a bi-modal distribution
630 of prey comprising both deep-sea squid and demersal fish (41). Smaller size squid (*Gonatus* sp.)
had a mean DML of approximately 225 mm, whereas the larger size squid (*Galiteuthis* sp.) was
about 625 mm in mean DML (41). We used the general scaling relationships between DML,
LRL, OL and weight to determine the average weights of typical and large squid prey (40, 41).
To estimate the energetic efficiency across the full distribution of prey types and sizes known for
635 *B. bairdii* (Table S9), we used the data from Walker et al. 2002 which includes family-specific
data for both deep sea cephalopods and fish. We note that we were unable to use the complete
data set from Ohizumi (2003) for this part of the analysis because they did not report family-
specific data for prey size.

640 1.8. Sperm whale (*Physeter macrocephalus*): Although there are multiple reports of giant squid
in the stomachs of sperm whales, the vast majority of captured prey includes relatively smaller
squid. For example, Clarke and Roper (62) showed that the dominant prey species were
Histioteuthis (78%), while the giant squid *Architeuthis* comprised only 0.12%. To explore the
impact of giant squid prey on foraging energetics in *P. macrocephalus*, we used the tag data of
645 Aoki et al. (2012), which revealed fewer prey capture attempts (mean= 1.5 feeding attempts per
dive) and higher escape speeds than the majority of tag data used in this study, in the context of

the inferred prey type (24 kg *Architeuthis*). Other larger prey comprises the following taxonomic groups: *Histioteuthis* spp., *Moroteuthis* spp., Ommastrephidae (*Dosidicus gigas*), Ommastrephidae (*Todarodes*). To calculate the energetic efficiency of foraging across a greater prey size distribution for sperm whales worldwide (Table S10), we used the combined family-specific data provide by Whitehead's Table 2.2 (63) and corresponding family-specific calorific values (64).

2. Rorqual whale prey data supplement

2.1. Data Collection: Acoustic backscatter data was collected using Simrad Ek60 or Ek80 transceivers at 38 and 120 kHz in three ecosystems: West Antarctic Peninsula (in humpback whale habitat in Andvord Bay and in minke whale habitat in Andvord and Paradise Bay), Southern California (SoCal) and Monterey Bay (Monterey). Ek80 transceivers were run in continuous wave (CW) mode. All systems were calibrated using a 38.1 mm tungsten carbide sphere (65) as close as possible to the time of data collection. All systems were calibrated either directly before or directly after the period of use. Upon analysis, however, the data from several calibrations were found to be insufficient, so later, more robust calibrations of the same system in identical configurations in the same ecosystem were applied to the data post-hoc even though they were further from the time of data collection. Specifically, for SoCal data a 2011 120 kHz calibration was also used in 2012, and a 2013 38 kHz calibration was used in 2012 and 2011. For Monterey data, three platform configurations were used to collect data. The RV John Martin has hull-mounted 38, 120 and 200 kHz transducers and an ek60 system. The 120 was calibrated

670 in 2016 and 2018, and the 38 and 200 were calibrated in 2018; all 2017 data were analyzed using
the 2018 calibration. Only data for which whales could be confirmed to be feeding on krill in the
vicinity of prey mapping were used, and data from 04.20.16 was excluded due to missing GPS
data.

675 2.2. Processing in Echoview v9:

The seafloor was detected by manually examining and adjusting an auto-generated line at least
0.5 m above the bottom. All data were visually inspected for anomalies such as false bottom
detections, noise spikes, or tracks of profiled instruments, and bad data at any frequency were
removed from all frequencies. In the Antarctic, backscatter from *Euphausia superba* were
680 delineated using dB-differencing as reported in Jarvis et al. (66). The threshold for analysis on all
echograms was set to -80 dB. A virtual echogram displayed the difference (subtraction) between
the 120 kHz and 38 kHz Sv (volume backscattering) echograms. Values of Sv(120-38) 2-16 dB
were colored red, indicating likely krill backscatter. School detection was performed using the
SHAPES algorithm (67, 68) and parameters were set to accept relatively small patches (< 40 m)
685 and link closely spaced patches (<5 m) in the 120kHz Sv echogram. Schools were manually
confirmed using discrete regions of the echogram to avoid the detection algorithm selecting
regions of high noise. In SoCal, schools were detected following Hazen et al. (69). For
detecting schools in Monterey, where krill were smaller than typically analyzed *E. superba*
resulting in higher differences between frequencies, school detection was applied to a 120-38
690 kHz differenced echogram so that only schools with greater than an 8 dB difference were
detected. Parameters used in school detection in Monterey were: minimum total school length =
10 m, minimum total school height = 3 m, minimum candidate length = 5 m, minimum candidate

height = 2 m, maximum vertical linking distance = 10 m, maximum horizontal distance = 50 m,
distance mode = GPS distance. Parameters used in school detection in Antarctica were:

695 minimum total school length = 10 m, minimum total school height = 5 m, minimum candidate
length = 5 m, minimum candidate height = 2 m, maximum vertical linking distance = 5 m,
maximum horizontal distance = 10 m, distance mode = GPS distance.

After detection, schools were adjusted manually to ensure that only appropriate krill schools
700 were detected. In Monterey, for instance, distributed scattering layers of low-density backscatter
(< -70 dB) made accurate identification of krill patches near the surface difficult using a threshold
of -80 . Higher thresholds (typically -62.25 dB, equivalent to the minimum threshold for blue
whale foraging on larger krill described below) were used on the 120 echogram to identify
patches within these layers.

705

Each echogram was then divided into cells representative of each species' gulp size by using the
average jaw length for the vertical cell size and the ventral groove blubber (VGB) length for the
horizontal cell size (67, 68). At the depths used, all patches had y-axis values larger than the
head width, so the extracted cells accurately represented a 2d projection of the gulp size. Minke
710 whales, had cells of 1.44 m (vertical) x 3.54 m (horizontal), humpbacks were 2.82 x 7.45, fins
were 3.95 x 11.29, blues were 5.02 x 14.71, and a theoretical big blue was 7.3 x 20.3. The mean
volume backscattering (MVBS) in dB re $1 \text{ m}^2\text{m}^{-3}$ of each cell within each school from the 120
kHz, 38 kHz, and a linearly subtracted 120-38 kHz echogram were then extracted for each study
species. The 120-38 linearly subtracted echogram was exported so that potential contributions
715 from siphonophores, at times abundant around Monterey Bay (69), could be extracted since

models suggest a flat frequency response curve for these gas-bearing organisms between 38 and 120 kHz (70). However, for consistency between ecosystems, only the 120 echogram was used to determine school biomass density. Additional dB analysis post-export is described below.

720 2.3. Krill patch biomass and energy determination: Three study areas, Andvord Bay, Antarctica (around humpback habitat), Paradise Bay, Antarctica (in minke habitat, also encompassing two surveys in icy habitat in Andvord Bay around foraging minke), and California (encompassing both the SoCal and Monterey ecosystems as blue, fin and humpback habitat) were processed using different representative krill populations.

725

For Antarctic ecosystems, acoustic surveys were complemented with net tows to determine the length-frequency distribution of krill (*E. superba*) targeted by tagged whales in the region. In minke habitat, krill distribution was bimodal with 61% of the catch of length $15.4 \text{ mm} \pm 2.9$ (mean \pm std. dev.) and 39% of the catch of length $39.1 \text{ mm} \pm 5.7$. In humpback habitat, krill was 730 $36.3 \text{ mm} \pm 6.2$. These distributions were used to calculate the mean target strength (TS) in each region using improved parameterization of the stochastic, distorted-wave Born-approximation with an $11^\circ \pm 4^\circ$ orientation distribution (71) as suggested by Jarvis et al. (66) to be most appropriate for Antarctic krill, and wet weight was calculated from length (66). These calculations resulted in TS of -78.4 dB and -81.2 dB and krill of 0.37 g and 0.19 g for humpback 735 habitat and minke habitat respectively.

For krill in California waters, two mean distributions were used, one for larger krill of primarily *T. spinifera* that make up over 80% of blue whale diets (72, 73) where length of ingested krill in

fecal samples was $19.3 \text{ mm} \pm 1.53$ (73), and one distribution of the smaller krill, *E. pacifica*,
740 which is abundant in the area but less targeted by whales, of $11.8 \text{ mm} \pm 3.32$ (73). Final results
of these two distributions were averaged for the energetic model. Since these krill are both
substantially smaller than *E. superba*, both in length and biomass/length, different TS and weight
metrics were used. TS were calculated using new calculations of the scattering properties of
small krill (J. Warren, personal communication): for the large krill (*T. spinifera*), TS was -93.2
745 dB; for small krill (*E. pacifica*), TS was -101.8 dB. Using a regression for *E. pacifica* and two
Thysanoessa sp. of $\log_{10}(W) = 3.119 \cdot \log_{10}(L) - 5.419$ (extracted from Fig 6 in 42) (W, weight
in mg; L, length in mm), the large krill distribution had mean biomass of 0.040 g/krill, and small
krill had mean biomass 0.011 g.

750 In addition to the dB differencing used to differentiate krill patches, each exported cell was
additionally subject to dB difference discrimination. Cells in schools were excluded from
analysis if the 120 kHz return signal was not 2-16 dB higher than the 38 kHz values (66) in the
Antarctic data. A cut-off of 10-24 dB was used in the California data, based on a suggested
differentiation of 14-20 dB (Warren, pers. com.), but extended to match the range of Antarctic
755 data so as not to overly exclude patches in one region.

Only prey patches in the depths examined in the study (deeper than two body lengths for minke
and one body length for other whales) were included in the prey analysis. Additionally, cells
were excluded if they were deeper than 225 m (below the depth of clean 120 kHz data), or if they
760 contained less than 49.5 g m^{-3} or more than 10 kg m^{-3} . This range was chosen to encapsulate
both the minimum threshold at which a 22 m blue whale would regain the energy expended

while foraging (71) up to a conservative estimate of the highest densities commonly reported in net tows (72). This upper threshold was specifically chosen as the point where the distributions of densities tailed off from a log-normal distribution with outliers that were likely a result of noise in the data that could not be filtered out or excluded. This threshold was 2.04-2.16 s.d. above the mean for all species using the big krill distribution (1.36-1.46 s.d. above mean for small krill distribution).

To calculate the energy per unit biomass of krill, for *E. superba* the calorimetric line from redoing the regression in (73) to force it through 0, and using a dry wt to wet wt conversion factor of 5 (74) gave a slope of 4575 kJ/kg. This is comparable to the value of 4645 used for *E. superba* by Clarke (75). For values more consistent with the results of the California krill, Chenowith (76) reported 3800 kJ/kg for krill of mean size 0.11 g, and for the small krill distribution, a value of 2940 kJ/kg for .07 g krill. Accordingly, 3800 kJ/kg was used for the large krill distribution and 2940 for the small krill.

2.4. Krill patch distribution and foraging bout scale analysis:

In total, between 51,569 and 262,815 feeding lunges (depending on species) were identified as meeting the criteria above. The overall distribution of gulp biomass was log-normal, so values were logged (to base 10) for all analyses below. To remove biases from differences in survey effort, each day of krill acoustic surveys was treated as an independent sample of krill distributions seen by foraging whales. If any individual day had less than 100 gulps, it was combined with other days of < 100 gulps in the same region to make a single sample. This left 4

785 days for each Antarctic krill survey, whereas surveys for other species had between 27 and 39
days of data. Patch distributions for each day were converted to density histograms, then we
assumed that whales forage on the densest regions they encounter during a dive. For this
analysis, each day was additionally divided into cells the length and height of the average
790 distance covered while foraging during a dive for each species (blue whales: 147 m x 40 m, fin
whales: 226 m x 57 m, humpback whales: 57 m x 24 m, minke whales: 109 m x 33 m) calculated
from georeferenced pseudo-tracks (i.e. dead reckoned reconstructed tracks) of foraging whales as
the mean vertical and horizontal extent of foraging from 10 s before the first lunge in a dive to 10
s after the last lunge in a dive. The maximum of the northing and easting extent of foraging was
used as the horizontal distance. The mean value of the top x number of gulp-sized patches
795 (where x is the median of the max number of lunges per dive for each species, blue whale = 5,
fin whale = 6, humpback whale = 7, minke whale = 9) in each dive-sized cell was calculated and
the distribution of these values for each day was determined as above.

800 **3. Prey data supplement for right whales and bowhead whales.**

We used data for prey density collected at the foraging grounds for both right whales (77, 78)
and bowhead whales (79). For bowhead whale prey data, we used an energy density of 26 kJ/g
dry weight (80), whereas for right whale prey data we used an energy density of 6.78 J/copepod
805 (81). We collated these data (right whales: Digitized data from figure 3a in Mayo & Marx, 1999,
and values from table 3 in Baumgartner & Mate, 2003; bowhead whales: mean *Calanus*
finmarchicus biomass values at foraging depths 75-115 m from figure 9 in Laidre et al. 2007 and

the standard deviation from Table 3 in Laidre et al. 2007) and estimated the distribution of prey densities that may occur on whale foraging grounds. These distributions were used as model input data to estimate the energetic efficiency of continuous ram filter feeding following the methods below in 5.3.

4. Tag data supplement: Using whale-borne tags to quantify foraging rates.

Suction-cup attached multi-sensor tags were used to collect data on diving and foraging performance (see Table S10, with additional information below). Inertial sensor data were analyzed using standard engineering techniques to determine whale orientation (pitch, roll, and heading) and fine-scale movement (82).

For toothed whales that use echolocation to find and capture prey, high-resolution digital acoustic recording tags (DTAG, www.soundtags.org/dtags)(82) were used to quantify prey capture attempts as high rate click trains (i.e. buzzes or creaks). The termination of buzzes and creaks in the acoustic data were analyzed in coordination with tri-axial inertial sensor data for peaks in the acceleration rate of change (jerk) (83), computed as the L2-norm of the jerk, which can be an indicator of prey capture (33, 84-86). However, reliable jerk signals associated with prey capture require tag placements that are sufficiently anterior on the whale's body. Therefore, all buzzes, irrespective of jerk signal, were used to compute the foraging rates. We only analyzed feeding attempt data at depths greater than 2 body lengths in shallow diving species (harbor porpoise and killer whale) and 3 body lengths in deep divers. We used stomach content analyses

of cephalopod beaks and fish otoliths from the scientific literature to estimate the distribution of prey types and sizes in foraging toothed whales (See section 1 above). When available, we used stomach content data that directly corresponded to the region where whales were tagged. If those data were not available, we used the next available data at a given spatial scale (i.e. ocean basin or global scale). We analyzed the frequency of prey types and sizes to estimate the amount of energy acquired by foraging whales across this prey range. We used values for cephalopod energy density (i.e. calorific value) obtained using adiabatic bomb calorimetry (64, 87). If species-specific or genera-specific values were not available, we either used values for closely related species as inferred from recent phylogenetic analyses (88) or median values for gelatinous/ammoniacal or muscular squid (64). For deep-sea fish (in our analysis, this applied only to prey for *B. bairdii*), we used the minimum values for demersal fish energy density (89) to reflect the deep foraging locations quantified by tag data (90).

Forrorqual whales, we used species-specific kinematic signatures that indicate lunge-feeding events (83, 91-93). In some species, we used Customized Animal Tracking Solutions (CATS, www.cats.is) equipped with 3D movement and video sensors to visually confirm lunge feeding events (92, 93). To complement these data sets, we also included DTAG deployments among several species. These kinematics signatures include an increase in speed driven by a bout of rapid fluke strokes (83, 94), the opening of the mouth at or less than one second from maximum speed (92), and a rapid deceleration phase after mouth opening (92, 95). Lunge feeding events are also frequently associated with rapid changes in body orientation (83, 92, 93, 96, 97), thus providing further evidence of prey capture events. We surveyed tag data to manually detect lunge-feeding events using all available evidence (i.e. video, speed, jerk). We only analyzed

lunge-feeding data when it occurred at depths greater than 1 body length of a typically sized
855 adult for the species (except for minke whales, where a threshold of 2 body lengths of a
typically-sized adult was used). For consistency in kinematic signatures(92) and energetic
efficiencies across species, we also only used tag deployments for rorqual whales foraging on
krill. Where available, krill feeding was confirmed with tag video and/or simultaneous prey
mapping with SIMRAD EK 60 and EK80 multi-frequency echosounders (14, 92, 93, 98-102).

860

Several tag data sets used in our analysis come from previously published studies (see also Table
S10), and details can be found therein for Risso's dolphin (85, 103, 104), Blainville's beaked
whale (44), harbor porpoise (33, 105) and southern resident killer whales (106-108).

865

4.1. Long-finned pilot whale (*Globicephala melas*): Foraging and dive data were obtained from
DTAG-2 (400-600 g) and DTAG-3 digital multi-sensor tags (82, 109) deployed on *G. melas* in
the Strait of Gibraltar (110, 111) over 3 field seasons in 2012 (DTAG-2), 2013 (DTAG-2,
DTAG-3), and 2015 (DTAG-3) (F. H. Jensen, unpublished data). DTAG-2 tags sampled 16-bit
stereo audio at 196 kHz (clip level of 171 dB re 1 μ Pa). The tags also contained pressure, tri-axial
870 acceleration, and tri-axial magnetic field sensors sampled at 50 Hz. DTAG-3 tags sampled 16-bit
stereo sound at 240 kHz (clip level of 179 dB re 1 μ Pa), while sampling the movement sensors at
250 Hz. All tags were deployed using an 8-m hand-held carbon fiber pole from a rigid-hulled
inflatable boat by slowly approaching or paralleling travelling groups of well-known, identified
individuals (112) from a small resident population. A single tagged individual per group was
875 used to avoid pseudoreplication. All work was completed under NMFS permit 14241 to Peter L.
Tyack.

880 4.2. Short-finned pilot whale (*Globicephala macrorhynchus*): Foraging and dive data were obtained from DTAG-3 tags (82) deployed in 2015 on *G. macrorhynchus* in the Azores, Portugal, using an 8-m hand-held carbon fiber pole from a 6.2-m rigid-hulled inflatable boat (113). The tags sampled 16-bit stereo audio at 240 kHz (clip level of 179 dB re 1 μ Pa) and pressure, tri-axial acceleration and tri-axial magnetic field at 200 Hz.

885 4.3. Cuvier's beaked whale (*Ziphius cavirostris*): Foraging and dive data were obtained from DTAG-3 tags (82) deployed in southern California (114) and off Terceira Island, the Azores, Portugal (115). The Azorean archipelago represents an open ocean, deep-water habitat where Cuvier's beaked whales are observed in relatively coastal waters (<7 nmi (Fleur Visser, unpublished data)). DTAG-3 tags were attached to the dorsal surface of the whales using an 8-m hand-held carbon fiber pole from a 6.2-m rigid-hulled inflatable boat. The tags sampled 16-bit
890 stereo audio at 240 kHz (clip level of 179 dB re 1 μ Pa) and pressure, tri-axial acceleration and tri-axial magnetic field at 200 Hz. All work was completed under permits issued by Secretaria Regional da Energia, Ambiente e Turismo of the Direção Regional do Ambiente (Horta, Faial, Açores, Portugal).

895 4.4. Baird's beaked whale (*Berardius bairdii*): Foraging data from one DTAG-3-tagged *B. bairdii* in southern California was obtained from a previous study (90). Because the tag fell off the whale during the ascent from a longer foraging dive of the data set, we included supplementary data, specifically post-dive surface interval, from another study of one whale tagged with a time-depth recorder (TDR) off the Pacific coast of Japan (116) to provide the post-

900 dive surface interval. The TDR tagged whale exhibited behavior very similar to the DTAG
tagged whale with foraging dives greater than 1000 m in depth. For the DTAG data set, we note
that acceleration (i.e. rate of acceleration, or “jerk”) from a tagged *B. bairdii* was much lower
during feeding events compared to steady swimming (90), so it was assumed that the whale
approached prey at relatively slower speeds in our energetics model.

905

4.5. Sperm whale (*Physeter macrocephalus*): Foraging data from DTAG-3 deployments off the
island of Dominica was collected as a part of a longitudinal behavioral study of well-known
social units (117). DTAG-3 tags were attached to the dorsal surface of the whales using an 9-m
hand-held carbon fiber pole from 11 m rigid-hulled inflatable boat. Tags sampled 16-bit stereo
910 audio at 120 or 125 kHz, providing a flat (± 2 dB) frequency response between 0.4 and 50 kHz,
and a clipping level of 184 dB re 1 μ Pa. Pressure and acceleration were sampled at a rate of 100
Hz and 500 Hz, respectively, both with 16-bit resolution. The whales were tagged under permits
P-122/4W-2, P-40/2W-7, and RP16-04/88FIS-9 issued by the Chief Fisheries Officer Mr.
Riviere Sebastien, Fisheries Division, Ministry of Agriculture and Fisheries, Government of
915 Dominica.

4.6. Antarctic minke whale (*Balaenoptera bonaerensis*): Foraging data from CATS tag
(Customized Animal Tracking Solutions; www.cats.is) deployments on *B. bonaerensis* were
obtained from the coastal waters of the Western Antarctic Peninsula during the austral
920 summer/autumn of 2018. CATS tags (approx. 685 g) were equipped with the following sensors:
pressure transducer (10 Hz), tri-axial accelerometers (400 Hz), magnetometers (50 Hz),
gyroscopes (50 Hz), fastloc GPS, 1080p up to 2k video and 12-48 kHz audio. Krill feeding was

confirmed with tag video and simultaneous prey mapping with SIMRAD EK60 and EK80 multi-frequency echosounders. Using methods similar to a previous publication (91), whales were
925 approached slowly with a 6-m aluminum hull boat equipped with an elevated pulpit and tagged using an 8-m carbon fiber pole. Tagging operations were performed under NMFS Permit 14809 and ACA permit 15-011.

4.7. Humpback whale (*Megaptera novaeangliae*): Foraging data from DTAG and CATS tag
930 deployments on *M. novaeangliae* were obtained from a wide range of locations, with details provided in several publications for DTAG deployments in Alaska and Antarctica (118-120) and for CATS tags in Cade et al. (2016). All remaining data sets were CATS tag deployments in several locations (Greenland; South Africa; Monterey, California; Stellwagen Bank National Marine Sanctuary). CATS tags were equipped with the following sensors: pressure transducer
935 (10 Hz), tri-axial accelerometers (400 Hz), magnetometers (50 Hz), gyroscopes (50 Hz), fastloc GPS, 1080p up to 2k video and 12-48 kHz audio. Tagging operations were performed under several NMFS permits (16111, 14809, 15271) through MULTI-2017-007.

4.8. Fin whale (*Balaenoptera physalus*): Foraging data from DTAG and CATS tag deployments
940 on *B. physalus* along the California coast (97, 121, 122), Greenland, and the Azores. DTAGs were equipped with the following sensors: pressure transducer (50 Hz), tri-axial accelerometers (250 Hz), magnetometers (50 Hz) and audio. CATS tags (approx. 685 g) were equipped with the following sensors: pressure transducer (10 Hz), tri-axial accelerometers (400 Hz), magnetometers (50 Hz), gyroscopes (50 Hz), fastloc GPS, 1080p up to 2k video and 12-48 kHz

945 audio. Tagging operations were performed under NMFS permits 16111 and 19116 and NMS
MULTI-2017-007.

4.9. Blue whale (*Balaenoptera musculus*): Foraging data from DTAG and CATS tag

950 deployments on *B. musculus* were obtained from several locations along the California coast
including Monterey Bay, Cordell Bank, San Diego, Channel Islands, and the greater southern
California Bight (92, 101, 121, 123). CATS tags were equipped with the following sensors:
pressure transducer (10 Hz), tri-axial accelerometers (400 Hz), magnetometers (50 Hz),
gyroscopes (50 Hz), fastloc GPS, 1080p up to 2k video and 12-48 kHz audio. Tagging operations
were performed under NMFS permits 16111 and 19116 and NMS MULTI-2017-007.

955

4.10. Bowhead whale (*Balaena mysticetus*) and North Atlantic Right Whale (*Eubalaena*

glacialis): Foraging and dive data recorded by DTAG tags (82) for *B. mysticetus* were obtained
in 2008 off Western Greenland (124), and for *E. glacialis* in the Bay of Fundy since 1999 (125,
126). Bowhead whales and right whales (Balaenidae) are continuous ram filter feeders and thus
960 require dynamic pressure generated from forward swimming to drive the filtration process (23,
127, 128). Although they sometimes exhibit periodic pauses (2 seconds in duration, at mean
intervals of 2.4 min) during the bottom phase of foraging dives (124, 129), it is not yet
demonstrated that these pauses indicate the culmination of a feeding bout or an engulfment
event. As demonstrated by Simon et al. (2009) and Van der Hoop et al. (2019), the kinematics
965 and foraging behavior of balaenids are characterized by near-continuous fluking and slow speeds
less than 1 m/s. Here we follow the inference made by Simon et al. (2009) that balaenid whales
“filter continuously throughout the bottom phase” of U-shaped foraging dives.

5. Bio-energetic modeling of foraging costs, gains, and efficiency

970 We used physics-based mechanical modeling to estimate the energy gained and used during foraging (16, 67, 68, 71, 96, 130, 131). Where possible, we used the kinematic data recorded by whale-borne tags to guide the modeling framework and estimate the forces at play during feeding events. This approach provides an estimate of the mechanical energy used for a given whale morphology within our current understanding of how whales feed (96, 132).

975 Mechanical energy was converted to metabolic energy using conservative loss coefficients. For toothed whales specifically, we modified a previously published unsteady hydromechanical model used for accelerating raptorial predators(133). For rorqual whales, we used unsteady hydromechanical models of engulfment (16, 67, 68, 71, 130). These energetic costs were added to baseline metabolic rate expenditures, which were estimated by
980 extrapolating allometric regressions from smaller mammalian species (68, 71, 134).

The bio-energetic modeling has been designed to reflect the salient physical characteristics of the three basic groups discussed in the paper: toothed whales (Odontoceti), bowhead and right whales (Balaenidae), and rorqual whales (Balaenopteridae). In trying to be as realistic as
985 possible, all models have also been tailored to the prey mobility and aggregation types that apply.

5.1 Toothed whale prey escape speeds

We used the relationship between maximum escape speed and fish body length
990 ($\log V_{\max} = 0.49 \log L^{0.60}$) provided by Domenici (11) to estimate the escape speed capability of

epipelagic fish targeted by tagged whales (relates only to the confirmed prey of *P. phoeocena*, *O. orca*). In contrast, demersal (i.e. benthic) fish and cephalopod prey were modeled to have much more limited effective escape speeds of 0.5 m s^{-1} due the constraints imposed by the sea floor (135); foraging close to the sea floor is suggested by the tag data in several toothed whale species (995) (*B. bairdii*, *M. densirostris*). Fish body length was estimated from echograms (33), through the age class analysis of fish scales from confirmed kills (56, 57), or from otoliths found in whale stomachs and regressions that relate otolith size to total fish length (41, 136). For general cephalopod prey less than 1 kg in body mass, we used the median maximum speed values (1.0 m s^{-1}) among cephalopod species (135). For odontocetes exhibiting high-speed pursuits during feeding events, such as *G. macrorhynchus* (50), we used the median maximum speed for pelagic cephalopods (2.0 m s^{-1}) as the input parameter for cephalopod escape speed (135). This escape speed value was also applied to any cephalopod prey that was larger than 1 kg such as giant squid (Architeuthidae) or large flying squid (*Todarodes* sp.). Where available, this resulted in maximum simulated predator speeds that were similar to those estimated from tag data. For example, computed and measured maximum speeds in *G. macrorhynchus* were 5.8 m s^{-1} and 6.0 m s^{-1} (50), respectively. Similarly, simulated and measured maximum speeds in *P. macrocephalus* were 3.3 m s^{-1} and 3.4 m s^{-1} (20), respectively. In contrast, some odontocetes (*M. densirostris*; *B. bairdii*) exhibit low speeds and dynamic accelerations during prey capture (43, 90); therefore, we selected 0.5 m s^{-1} as the input parameter for cephalopod or deep sea fish escape speed. Such diminished escape performance capacity may be due to inhabiting extreme low oxygen waters (137). This resulted in maximum simulated predator speeds that were similar typical cruising speeds ($2\text{-}3 \text{ m s}^{-1}$) in large breath-hold divers (138). Two exceptions were *O. orca* and *P. phocoena* (simulated speeds of 5.3 m s^{-1} and 4.8 m s^{-1} , respectively), which were

actively pursuing epipelagic fish with more developed escape responses and inhabit relatively well-lit environments. Using an echogram scoring procedure, Wisniewska (33) estimated a 94% success rate for prey capture in harbor porpoises (*Phocoena phocoena*). We thus assumed a 100% success rate among all whales assuming that capture efficiency is invariant of predator mass.

5.2. Toothed whales capturing single prey (cephalopods or fish): A simple model of raptorial energy expenditures during prey approach and capture can be devised by considering the energy spent by a predator accelerating along a straight line towards a single prey item (i.e. fish or squid), which escapes in a direction orthogonal to the approaching whale, and at a constant swim speed (V_{prey}). Here the prey begins its escape upon sensing the approaching predator a distance RD away (the so-called “reaction distance”). With most toothed whales in this study capturing prey during deep foraging dives in extremely low light habitats, we assumed $RD=0.05L_{body}$ in all cases. In this scenario the prey can avoid capture only if it can get “out of the way” of the approaching whale by the time T_{capt} at which the prey finds itself just outside the predator’s maximum skull width (generally located at the temporo-mandibular joint) (67), here approximated as the width of the antorbital notches or antorbital process of maxillae (139). Orthogonal escapes are somewhat idealized, whereas visually mediated escape sprints tend to occur at angles greater than 90° (140). However, this assumption should be a sufficient proxy for prey escape strategies that require maximal expenditures from the predator.

This calculation is based on two components, namely, the mathematical representation of the prey and predator kinematics; and a calculation of the mechanical propulsive work carried out by

the heaving caudal tail during the predator's accelerative stage (i.e., from $t = 0$ to T_{capt}). During a constant-acceleration regime, the predator's kinematics can be completely specified by the following inputs: first, the prey's reaction distance RD and its escape speed V_{fish} ; and secondly, the predator's maximal skull width (w_{max}) and initial speed V_i when the prey senses the approaching predator. From these the predator's acceleration a , mean speed $\langle V \rangle$, accelerative stage duration T_{capt} and speed V_f at T_{capt} , are derived and calculated as follows:

$$\begin{aligned}\langle V \rangle &= 2V_{prey} \frac{RD}{w_{max}} \\ T_{capt} &= \frac{w_{max}}{2V_{prey}} \\ a &= 2 \frac{\langle V \rangle^2}{RD} \left(1 - \frac{V_i}{\langle V \rangle} \right) \\ V_f &= aT_{capt} + V_i\end{aligned}\tag{So-1}$$

The mechanical energy and power expended by the predator's fluking are calculated in a first step from the work-energy theorem, itself derived from integrating Newton's 2nd law of motion over the distance traveled (67, 141):

$$\frac{1}{2} M_{body} (V_f^2 - V_i^2) = \Delta W_{flukes} - \Delta W_{drag}\tag{So-2}$$

The left-hand-side (LHS) corresponds to the predator's change in kinetic energy ΔKE_{body} , with M_{body} corresponding to the predator's body mass (a known quantity here); and the right-hand-side (RHS), to the work (ΔW_{flukes}) performed by the fluke's propulsive force, minus that of the

drag force (ΔW_{drag}). Eq. So-2 thus allows for the calculation of the propulsive mechanical work via $\Delta W_{flukes} = \Delta KE_{body} + \Delta W_{drag}$, i.e., from the prior calculation of the kinetic energy change via Eqs. So-1, plus that of the drag explained next.

1060 The drag modeling follows an approach used recently in the context of accelerating sharks (133). It is based on splitting the drag force into a so-called “parasite” component (D_{para}), which parameterizes the energy losses due to body viscous friction against the surrounding water, and resulting turbulent near-wake (142, 143); and the acceleration reaction component (D_{ar}), which accounts for the energy transferred to the accelerating water enveloping the body and associated
1065 boundary layer (133, 144). (The latter is sometimes known as “potential flow” (145)):

$$\begin{aligned}
 D &= D_{para} + D_{ar} \\
 D_{para} &= \gamma_{tail} \gamma_{depth} \frac{1}{2} \rho_{water} S_{wet} C_D(t) V^2(t) \\
 S_{wet} &= 0.08 M_{body}^{0.65} \\
 C_D(t) &= \frac{0.072}{(R_e)^{1/5}} \left[1 + 1.5 \left(\frac{w_{max}}{L_{body}} \right)^{3/2} + 7.0 \left(\frac{w_{max}}{L_{body}} \right)^3 \right] \quad (\text{So-3}) \\
 R_e &= \frac{V(t) L_{body}}{\nu} \\
 D_{ar} &= \gamma_{tail} \gamma_{depth} k M_{body} a
 \end{aligned}$$

Eqs. So-3 involve the sea water mass density ρ_{water} ($= 1027 \text{ kg m}^{-3}$) and kinematic viscosity ν ($= 1.15 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$), predator body wetted area S_{wet} (146, 147), and acceleration reaction (or “added mass”) coefficient k . The latter is calculated at values ranging from 0.05 to 0.08, depending on body fineness ratio, from a formula derived by Lamb (1932) for ellipsoids of

revolution. (The formula can be found also in the Supplement in Gleiss et al. 2017). Finally, the coefficient γ_{tail} corresponds to a correction for the heaving tail which generates drag in addition to the rigid-body drag described by Eq. So-3. The value of γ_{tail} is estimated at ~ 2 to 3 from a comparison of rigid body drag and fluking thrust estimated using lunate tail fluid dynamics (146-148). The factor γ_{depth} parameterizes the increase of drag at the surface (wave drag) (149, 150) and underneath it (blockage drag) (149). Generally, $\gamma_{\text{depth}} \sim 1$ (as done here) at depth and ~ 3 -5 very close to the surface (and at high speed).

The parasite drag coefficient C_D above assumes a turbulent boundary covering the entire body, a valid assumption given the high Reynolds (R_e) number characterizing the flows of odontocete accelerative stages ($R_e > 10^6$, with $V_i \sim 1\text{m/s}$). This is a range in which there is no laminar-turbulent transition of the boundary layer. Additionally, the use of Eqs. So-3 assumes that no flow separation occurs over time – again a reasonable assumption here given the extreme taper of the body and the high Reynolds number at play (146, 151). (Flow separation and attendant large unsteady drag effects appear in accelerations of bluff bodies starting from a state of rest (152)).

The drag work that follows is first integrated,

$$\Delta W_{\text{drag}} = \int D(t)dx = \int D(t) \frac{dx}{dt} dt = \int_0^{T_{\text{capt}}} D(t)V(t)dt \quad , \quad (\text{So-4})$$

and then calculated exactly given the profile $V(t) = at + V_i$ and Eq. So-3. From this result, and using $\Delta W_{\text{flukes}} = \Delta KE_{\text{body}} + \Delta W_{\text{drag}}$, one obtains the (propulsive) mechanical energy expended by

the flukes. After including metabolic and propeller efficiency coefficients ($\eta_{metab} = 0.25$ and $\eta_{prop} \sim 0.7-0.8$) (147), as well as adding the metabolic expenditures of the rest of the body (i.e., those not directly associated with its locomotor structures), the metabolic rate of fluking expenditures is finally calculated as follows

1100

$$\begin{aligned} \text{Fluking MR} = & 6.72 M_{body}^{0.68} + \\ & \frac{1}{\eta_{metab}\eta_{prop}} \left\{ \frac{1}{T_{capt}} \frac{M_{body}}{2} \left[(aT_{capt} + V_i)^2 - V_i^2 \right] + \gamma_{tail}\gamma_{depth} kM_{body} a \left(\frac{aT_{capt}}{2} + V_i \right) \right\} + \\ & \frac{1}{\eta_{metab}\eta_{prop}} \left\{ \gamma_{tail}\gamma_{depth} \frac{1}{2} \rho_{water} S_{wet} \frac{0.072}{(R_e)^{1/5}} \left[1 + 1.5 \left(\frac{w_{max}}{L_{body}} \right)^{3/2} + 7.0 \left(\frac{w_{max}}{L_{body}} \right)^3 \right] \left(\frac{V_i^{1/5}}{3.8aT_{capt}} \right) \left[(aT_{capt} + V_i)^{3.8} - V_i^{3.8} \right] \right\} \end{aligned} \quad (\text{So-5})$$

The rest-of-body expenditures formula used here is the marine mammal resting metabolic rate (134, 153). This metabolic rate was also applied to all periods of time associated with non-foraging dive phases (i.e. descent, ascent, time in-between foraging events) and post-dive surface time.

1105

1110

5.3. Bowhead whales and right whales capturing copepod aggregations using continuous ram filtration: Balaenid whales are so-called continuous filter-feeders which exploit limited-mobility prey such as copepods and krill that live in aggregations of size greater than, or similar to the whales' size. Prey capture (in the bulk) is accomplished by swimming through an aggregation

with their mouth open, thereby exposing their baleen to the oncoming prey-water mixture (96),
 1115 with the latter used as a filter to retain the prey (127, 154).

The low-mobility of the prey allows the calculation of the energy gained by viewing the prey and
 water ahead as being “swept” into the mouth, with the whale swimming through a patch at
 constant speed. Thus the rate of prey items captured is calculated by multiplying the flux of
 1120 swept-in water by the whale’s speed (U), and the energy E_{coll} gained (and assimilated) obtained
 by multiplying the latter by prey volumetric density (CPCM; item per m^3), the energetic density
 of a single prey item (EPC ; Joules/prey item), and foraging times (T_{forage}) (77):

$$E_{coll} \Big|_{balaenid} = 0.84 \cdot EPC \cdot h_{HT} \cdot 2D_{in} \cdot CPCM \cdot U \cdot T_{forage} \quad (\text{Sb-1})$$

1125 Parameters h_{HT} and $2D_{in}$ correspond to the mouth inlet height and width respectively (23).
 Coefficient “0.84” is the percentage of energy obtained after assimilation of the prey (155).

The energy overhead expended to capture the prey is based on adding the metabolic energy
 1130 ($E_{drag}^{balaenid}$) expended by the locomotor musculature - a velocity-dependent term which will be
 equated to drag-, plus that expended by the rest of the body ($robMR \times T_{forage}$), a velocity-
 independent term (156, 157).

$$E_{expd} \Big|_{balaenid} = robMR \Big|_{balaenid} T_{forage} + E_{drag}^{balaenid} \quad (\text{Sb-2})$$

The factor $robMR$ is a metabolic rate (Watts) based on the resting metabolic rates of marine mammals (134, 153).

$$robMR|_{balaenid} = 0.8 \times 6.72 M_c^{0.68} \quad (Sb-3)$$

1140 The factor “0.8” has been added to account for balaenids being grazers rather than raptors, a correction factor estimated from the ratio of the basal metabolic rates by Kleiber (158) for land grazers over the values provided by Hemmingsen (159) which included both land herbivores and carnivores.

1145 The energy expended to compensate for drag losses in balaenids is similar to other approaches (160, 161), and is given by:

$$E_{drag}^{balaenid} = \frac{1}{2} \rho_w S_{wetted} C_D^{balaenid} U^3 \frac{\eta_{heave}}{\eta_{metab} \eta_{prop}} T_{forage} \quad (Sb-4)$$

1150 The η 's correspond to efficiencies: “metabolic” η_{metab} (= 0.25) and “propeller” η_{prop} ($\sim 0.7 - 0.8$) (146, 147); and “tail-heaving” η_{heave} ($\sim 2 - 3$) to account for the drag of the heaving tail when C_D is calculated from a formula validated with rigid-body hydrodynamics (water and wind tunnels, and the like). (Parameter η_{heave} is the ratio of the drag of a tail-heaving swimmer, over the drag of the same swimmer but in a rigid body configuration (146, 162). Parameters ρ_{water} , C_D and S_{wetted} are density of sea water, the body drag coefficient (mouth open) (23) and the body wetted body surface area (mouth closed) (146, 163) respectively:

1155

$$C_D^{\text{balaenid}} = 0.0060 + 0.0084A_{in} + 0.0055A_{in}^2 \quad (\text{Sb-5})$$

1160
$$S_{\text{wetted}} = 0.08M_c^{0.65} \quad (\text{Sb-6})$$

In Eq. Sb-5 parameter A_{in} is the half-area of the mouth inlet ($= D_{in} h_{HT}$; (23)) and in Eq. S-6, M_c the body mass, here calculated from a correlation based on body length (L_{body}) (164):

1165
$$M_c = 13.2L_{body}^{3.06} \quad (\text{Sb-7})$$

5.4. Rorqual whales capturing aggregations of krill: Rorqual whales are intermittent ram filter-feeders, or lunge filter feeders, in which the stages of prey approach, prey capture (via
1170 engulfment) and prey retention (via engulfed water filtration through baleen) are carried out sequentially, in contrast with the balaenids which carry out those tasks simultaneously using continuous ram filtration (96).

More specifically (83, 94), rorquals approach large prey aggregations (patches of krill and/or
1175 schools of fish) at high-speeds while fluking heavily (i.e., accelerating to speeds well exceeding 2m s^{-1}). This stage is followed by the engulfment of the prey and water in which it is embedded, a process aided by the distension of the VGB, which then doubles (or more) overall body mass (165, 166). Most importantly for the modeling, this engulfment stage is executed with little
1180 fluking, thereby leading to a decelerative stage in which the VGB's own push onto the prey-laden water generates in reaction a different form of drag – engulfment drag (130). Finally,

filtration through baleen of the engulfed mass occurs at slower speeds (at speeds $< 1\text{ m s}^{-1}$) and over times that are typically at least 10 times longer than those associated with prey approach and engulfment (16, 167). Herein foraging expenditure calculations proceed in a manner similar to the previous odontocete case, i.e., via the use of the work-energy theorem (Eqs. So-2 and So-4), but with approximations specific to each of the three stages of rorqual lunge-feeding (16, 67, 71).

Two such approximations arise during engulfment. The first concerns the amount of engulfed mass, assumed here as obtained at maximum gape ($\sim 80^\circ$), and of volumes approximated by two quarter-ellipsoid sections (168):

$$M_{\text{water}} = \rho_{\text{water}} \cdot \frac{\pi}{3} 1.17 L_{\text{VGB}} L_{\text{mandible}} \frac{1}{2} w_{\text{skull}} \quad (\text{Sr-1})$$

Parameters L_{VGB} , L_{mandible} and w_{skull} correspond to the longitudinal length of the filled ventral pouch (a.k.a. VGB), the (longitudinal) length of the mandibles and maximal width of the skull (measured near the mandible hinge line).

Another important approximation is the whale's speed U_{close} at the end of engulfment (and mouth closure) which, at least for the maximal lunges towards krill, can be related to the speed U_{open} at the onset of engulfment (mouth opening) as follows:

$$U_{\text{close}} = U_{\text{open}} \left(\frac{M_{\text{whale}}}{M_{\text{whale}} + M_{\text{water}}} \right) \quad (\text{Sr-2})$$

1205 This result arises from applying momentum conservation while viewing engulfment as a perfectly inelastic collision between a rorqual and its to-be-engulfed mass. Equation Sr-2 works well when compared with an average of all tag data collected during krill-feeding lunges at depth and performed during many dives, namely $M_{whale}/(M_{whale} + M_{water}) \sim 0.42 - 0.47$ (calculated) versus $U_f/U_i = 0.43 \pm 0.03$ (tag data) (92). This approximation, along with the values of U_{open} obtained from tag data, allows for the full computation of the work ΔW_{flukes} and $\Delta W_{VGBdrag}$ generated.

The mechanical work of relevance, i.e., that of the muscularized VGB during engulfment ($\Delta W_{VGBdrag}$) or fluking caudal tail during prey approach (ΔW_{flukes}), is obtained again for the work-energy theorem (16, 67, 71):

1215

$$\text{Prey-approach: } \frac{1}{2}(M_{body} + M_{ar})(U_{open}^2 - U_{close}^2) = \Delta W_{flukes} - \Delta W_{drag} \sim \Delta W_{flukes} \quad (\text{Sr-3})$$

$$\text{Engulfment: } \frac{1}{2}(M_{body} + M_{ar})(U_{close}^2 - U_{open}^2) = \Delta W_{flukes} - \Delta W_{drag} \sim -\Delta W_{drag} \quad (\text{Sr-4})$$

1220

with parameter M_{ar} corresponding to an acceleration reaction mass scale defined by $M_{ar} = k M_{body}$, and parameter k calculated again from Lamb's analysis of acceleration reaction drag of ellipsoid of revolution (133) which for rorquals ~ 0.03 (Balaenopteridae) and ~ 0.045 (*Megaptera*). Comparing Eqs. Sr-3 and Sr-4 shows the same speeds U_{open} and U_{close} (where $U_{open} > U_{close}$) being used interchangeably, a result of assuming the whale's speed at the beginning of

1225 prey approach being very similar to the speed at the end of engulfment – a behavior repeatedly
confirmed by tag data averaged over dozens of dives and lunges (92).

We note also that in Eq. Sr-3, the drag work term was neglected as a result of being very small in
comparison to the increase in kinetic energy - for example ~10x smaller for a 25m blue whale
1230 accelerating from 1.7 to 3.75 m/s. More generally-speaking, the approximation should be
sufficiently accurate at $M_c > 30,000\text{kg}$ ($\log M_c > 4.5$), i.e., the mass range shown in Extended
Figure 2 that includes the peak and efficiency drop-off of the efficiency. Below 30,000kg, and
particularly for the smaller rorquals such as the minke whales, this drag term increases the
expenditures by about 30-40%, thereby reducing the efficiency significantly.

1235
In Eq. Sr-4, on the other hand, neglecting the fluking work term was motivated by the near-
absence of fluking in surface lunge feeding videos (and at depth, as documented by archival
video and 3D motion sensing tags by Cade et al (2016)). Also neglected was the so-called
“shape” drag caused by the flows that travel *around* the body (130), which is generally much
1240 smaller than the “engulfment” drag associated with the flows that end up into the buccal cavity
(16). The “drag work” that remains is mostly that of the muscularized VGB which pushes the
engulfed mass forward (during engulfment) in order to set it into motion and at the speed of the
whale upon mouth closure (67, 68, 130).

1245 As in the modeling scenario for odontocetes, the metabolic expenditures (total metabolic rate)
are obtained by first adding the mechanical work found on the RHS of Eqs. Sr-3 and Sr-4. The
result is then corrected for metabolic and propeller efficiencies ($\eta_{\text{metab}} = 0.25$ and $\eta_{\text{prop}} \sim 0.7-0.8$)

(147), and lastly added to the metabolic expenditures of the rest of the body during the three stages of a lunge:

1250

$$\text{Lunge \& Purge MR} = f \cdot (4.1M^{0.75}) + 2 \cdot \frac{1}{2} (M_{body} + M_{ar}) U_{open}^2 \left[1 - \left(\frac{M_{body}}{M_{body} + M_{water}} \right)^2 \right] \left(\frac{1}{\eta_{metab} \eta_{prop}} \right) \left(\frac{1}{T_{approach} + T_{engulf} + T_{purge\&filter}} \right) \quad (\text{Sr-5})$$

Parameters $T_{approach}$, T_{engulf} and $T_{purge\&filter}$ correspond to the durations of the prey patch approach and engulfment, and water expulsion respectively. The first term on the RHS represents the rest-of-body metabolic rate, here estimated from Hemmingsen's land vertebrates (1960) Basal Metabolic Rate formula and corrected by factor f (~ 1 to 2) to account for the likely higher level of metabolic activity in rorquals during multi-lunge dives (155, 169). The second term incorporates the rate of work of the relevant locomotor appendages during prey-approach and VGB during engulfment, but it omits that performed by VGB musculature during purging and filtration, on account of the very long durations involved.

1260

With respect to the prey energy acquired, and during a so-called "maximal" lunge in which the maximum gape reaches approximately 80° , where the buccal cavity should inflate to maximum capacity (166, 168), calculating the energy extracted from the captured prey proceeds as follows:

1265

$$E_{coll} |_{\text{balaenid}} = 0.84 \cdot \left(\frac{M_{water}}{\rho_{water}} \right) \cdot \frac{\text{Joules}}{\text{krill}} \cdot \frac{\text{krill}}{\text{m}^3} \quad , \quad (\text{Sr-6})$$

with the factor M_{water} calculated from Eq. Sr-1. As with balaenids, an assumption is made on the inability by the prey to swim fast enough and in a coordinated way, to affect a significant escape; hence the absence of a prey collection efficiency in the formula.

6. Statistical Analyses

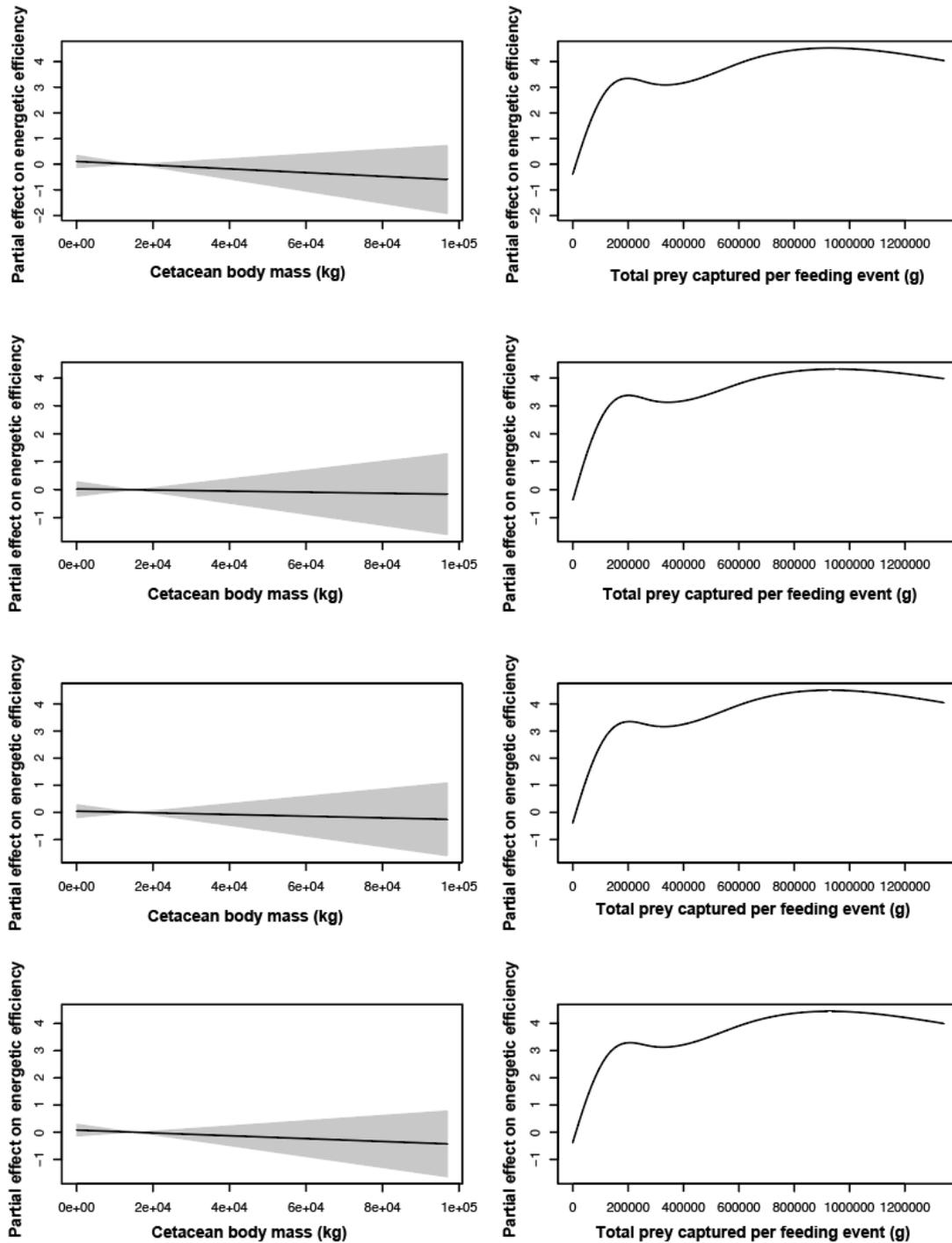
We built a series of linear models to explore the relationships between cetacean diving and foraging capacities and body size (see the R Markdown files in the Supplementary Material Part 2). All models included a categorical covariate for filter feeders and single-prey feeders to test if the fundamental relationships were different between the two guilds. Since information on prey and body size was not available for each tagged whale in this study, but rather the population or species as a whole, the models were run on species-specific means. Prey energy data were weighted by the relative frequency of occurrence of prey categories (defined by prey species and/or size, see Tables S2-S9) based on stomach contents data and prey mapping data for toothed whales and baleen whales, respectively. Given the paucity of prey data by species, the 95% confidence intervals for model parameters and 95% confidence bands for model fits were estimated from 10,000 bootstrap replicates on prey categories (i.e. 10,000 foraging scenarios) using the bias-corrected and accelerated and the percentile methods, respectively (170). The data were analyzed in a phylogenetic context to account for their non-independence due to shared evolutionary history of the species (171). To that end, we conducted a phylogenetic Generalized Least Squares analysis (pGLS) (172) using the *gls* function in the *nlme* package in R version 3.6.1 (173), and the cetacean phylogenetic tree published in McGowen et al. (174). The

tree was downloaded from TreeBase (175) and edited in Mesquite (176) to remove species with no data on foraging performance, and to add timing of diversification of branches based on the results of McGowen et al. (174). Phylogenetic information was incorporated into the models using Pagel's λ correlation structure (177), constructed with the *corPagel* function in the *ape* package (178), with the amount of phylogenetic signal in the data estimated using restricted maximum likelihood (REML) approach. $\lambda = 0$ suggests that the relationship between predictor and response variables is unrelated to phylogeny, while $\lambda = 1$ indicates that traits have evolved under Brownian motion on the given phylogeny. Intermediate values of λ indicate that traits have evolved according to a process in which the effect of phylogeny is weaker than in the Brownian model, while values of $\lambda > 1$ can arise if, for instance, traits are more similar than predicted by Brownian motion, given the phylogeny (179). When the latter case occurred, λ was fixed at 1. When λ was negative, suggesting that closely related species have negatively correlated phenotypes under the Brownian model of evolution, λ was fixed at 0.

We acknowledge that feeding mode (treated as a covariate in the models) is tightly linked to phylogeny. However, Pagel's λ represents the average phylogenetic signal across the entire tree, and we expected to have more regional phylogenetic signal in the tree (for example for beaked whales within the odontocetes). To test whether the inclusion of both the feeding mode covariate and the phylogenetic correlation structure in the model removes power from either of the two, for each of our five datasets (i.e. prey energy-body mass (Figure 3) and energetic efficiency-body mass for four metabolic rate scaling assumptions (Figure 4)), we also estimated Pagel's λ for a model without the covariate and ran an ordinary least squares (OLS) model that included the covariate, but not the correlation structure (see the R Markdown files in the Supplementary

Material Part 2). For all five datasets, Pagel's λ of the full model (i.e. pGLS with the covariate)
1315 was high and similar to that of the pGLS without the covariate (~ 1). At the same time, the
covariate was significant in the full pGLS, and the model remained informative. While ordering
the residuals of the OLS by phylogeny suggested presence of phylogeny-based correlation in the
data (supporting the need for a pGLS, even when the feeding mode covariate is included), the
plot of autocorrelation function (*acf* in R) on those residuals did not exhibit significant spikes,
1320 likely because of the limited number of species in the data. The low number of species included
in the best available dataset, however, also limits our phylogenetic inference in that we could not
test competing models of evolutionary processes. Furthermore, we could plot confidence bands
with OLS fits, but not pGLS fits (180). Therefore, while results of both pGLS and OLS are
provided in Table S11 and the R Markdown file, we present the results of the OLS in the figures
1325 of the main manuscript.

Standard methods, uninformed by phylogenetic relationships, were also used to further test the
hypotheses in this study. We ran a generalized additive mixed model (181) (in *mgcv* v1.8-21)
with species as a random covariate, to further explore and model the non-linear effects of body
1330 size and foraging capacity on energetics. We fitted the model using a Poisson family with a log
link and limited the "knots" to 5 to reduce overfitting. This equation took the general form: $E_E =$
 $f(\text{body mass}) + f(\text{feeding capacity}) + \varepsilon(\text{species})$. We used a non-parametric bootstrap given the
paucity of prey data by species to create 100 foraging scenarios per species.



1335

Figure S1. Generalized additive mixed model partial plots for the effect of (baleen and toothed) whale body size and prey intake (for either feeding mode) on the energetic efficiency of a foraging bout. Partial plots represent the effect of each modeled component. In

this case, and across difference scaling trajectories of metabolic rate (top to bottom panels: 0.45,
1340 0.61, 0.68, 0.75), efficiency is driven by the ability to forage on large, high calorie individual
prey items (odontocetes) or dense prey patches (mysticetes) rather than the metabolic advantages
associated with large body size.

1345

1350

1355

1360

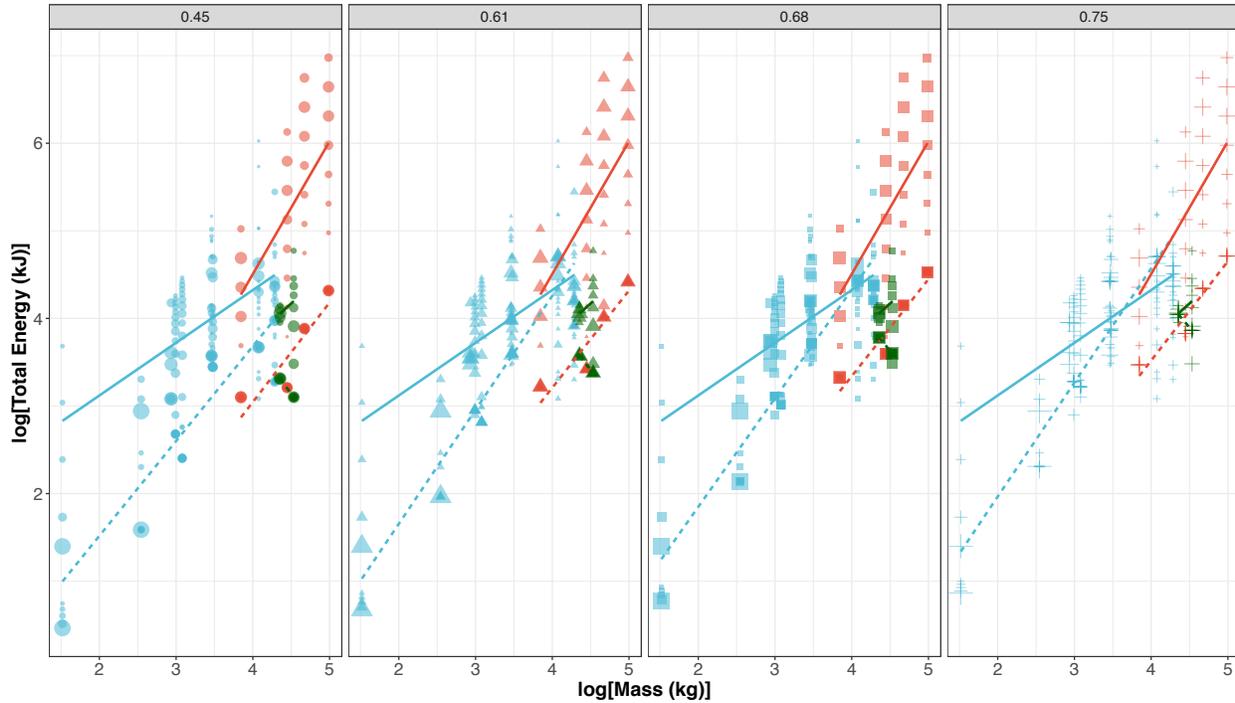


Figure S2. Scaling of energetic gains (solid lines, lighter symbols) and costs (dashed lines, darker symbols) for foraging dives and corresponding surface intervals in toothed whales

1365

(blue), rorqual whales (red), and balaenid whales (green). Assumptions about the scaling of metabolic rate (plus symbols, $MR \propto Mc^{0.75}$; squares, $MR \propto Mc^{0.68}$; triangles, $MR \propto Mc^{0.61}$; circles, $MR \propto Mc^{0.45}$) change the energy expenditure during dives relative to energy gains that were determined from empirically measured feeding rates (Fig. 2) and distributions of prey quality (Fig. 3). The ratio of energy gain to energy cost determines the energetic efficiency of foraging

1370

(Fig. 4). Linear regressions are shown for each group and each energy flux direction. The vertical spread of the data corresponds to prey quality distribution data (as in Fig. 3), with larger icons denoting greater proportions of observed values.

1375

1380

<i>Fish size</i>	hp12_272a	hp13_102a	hp12_293a	hp14_226b
<5 cm	29	29	22	16
5-10 cm	1	1	5	10
10-15 cm	0	0	2	4
15-20 cm	0	0	2	1
20-25 cm	0	0	1	0

Table S1. Estimated lengths of fish targeted by tagged *P. phocoena* (33).

1385

<i>W</i> (g)	Energy density (kJ/g)	Total energy (kJ)	Percent of diet
2.39	4.4	10.5	78.0
4.74	4.8	22.7	13.8
18.64	5.6	103.4	4.9
73.37	6.3	462.3	2.4
288.74	7.1	2037.3	0.8

Table S2. Calculated weights, energy density and total energy content of fish (*Clupea harengus*) targeted by tagged *P. phocoena* for energetics modeling.

Family	Genus	Species	W (g)	E (kJ/g)	E (kJ)	Percent of diet
Onychoteuthidae	<i>Ancistroteuthis</i>	<i>lichtensteini</i>	164.1	2.31	379.0	9.1
Histioteuthidae	<i>Histioteuthis</i>	<i>bonnellii</i>	24.1	2.65	63.7	3.6
Histioteuthidae	<i>Histioteuthis</i>	<i>reversa</i>	104.5	2.65	276.8	78.2
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	159.6	4.01	639.9	7.3
Heteroteuthidae	<i>Heteroteuthis</i>	<i>dispar</i>	34.8	2.65	92.2	1.8

Table S3. Prey size distribution and calorific value for *G. griseus* energetics modeling.

Family	Genus	Species	W (g)	E (kJ/g)	E (kJ)	Percent of diet
Cranchiidae	<i>Taonius</i>	<i>pavo</i>	122.2	1.69	207	42.9
Histioteuthidae	<i>Histioteuthis</i>	<i>reversa</i>	57.2	2.65	156	42.9
Octopoteuthidae	<i>Octopoteuthis</i>	<i>sicula</i>	247.8	3.08	763	14.3

1390

Table S4. Prey size distribution and calorific value for *M. densirostris* energetics modeling.

Family	Genus	Species	W (g)	E (kJ/g)	E (kJ)	Percent of diet
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	50.0	4.01	201	10.0
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	150.0	4.01	602	17.3
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	250.0	4.01	1003	12.0
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	350	4.01	1404	7.3
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	450	4.01	1805	13.3
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	550	4.01	2206	13.3
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	650	4.01	2607	10.0
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	750	4.01	3008	4.7
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	850	4.01	3409	4.7
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	950	4.01	3810	4.7
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	1050	4.01	4211	2.0
Ommastrephidae	<i>Todarodes</i>	<i>sagittatus</i>	1150	4.01	4612	0.7

Table S5. Prey size distribution and calorific value for *Globicephala* energetics modeling.

1395

Family	W (g)	E (kJ/g)	E (kJ)	Percent of diet
Enoploteuthidae	46	2.31	106	0.005
Ancistrocheiridae	1183	2.31	2733	0.016
Octopoteuthidae	154	3.08	474	16.8
Onychoteuthidae	631	4.53	2859	0.14
Gonatidae	259	3.78	1197	30.5
Histioteuthidae	64	2.65	490	4.8
Ommastrephidae	1675	4.01	3766	1.2
Pholidoteuthidae	2257	3.78	3134	0.78
Lepidoteuthidae	1729	3.08	5325	0.011
Cycloteuthidae	802	2.31	1853	0.022
Chiroteuthidae	221	2.31	510	0.56
Mastigoteuthidae	272	1.82	183	4.2
Cranchiidae	79	1.69	138	32.3
Vampyroteuthidae	181	2.31	418	8.8
Opisthoteuthidae	87	3.77	328	0.005
Bolitaeneidae	65	3.77	244	0.011

Table S6. Prey size distribution and calorific value for *Z. cavirostris* energetics modeling.

1400

<i>Genus species</i>	W (g)	Energy density (kJ/g)	Total energy (kJ)	Percent of diet
<i>Oncorhynchus tshawytscha</i>	1100	5.63	6192	1.9
<i>Oncorhynchus tshawytscha</i>	3100	5.74	17782	4.7
<i>Oncorhynchus tshawytscha</i>	8500	5.86	49781	21.7
<i>Oncorhynchus tshawytscha</i>	13300	5.98	79573	18.85
<i>Oncorhynchus tshawytscha</i>	13700	6.24	85539	2.85
<i>Oncorhynchus kisutch</i>	1138	4.58	5212	0.5
<i>Oncorhynchus kisutch</i>	1703	4.57	7782	2
<i>Oncorhynchus kisutch</i>	2267	4.57	10351	12.5
<i>Oncorhynchus kisutch</i>	2832	4.56	12920	20
<i>Oncorhynchus kisutch</i>	3397	4.56	15489	12.5
<i>Oncorhynchus kisutch</i>	3961	4.56	18058	2
<i>Oncorhynchus kisutch</i>	4526	4.56	20627	0.5

Table S7. Fish (*Oncorhynchus sp.*) weight, energy density and total energy content used for *O. orca* energetics modeling.

1405

1410

1415

Family	W (g)	Energy density (kJ/g)	Total energy (kJ)	Percent of diet
Moridae	915	1.7	1,532	36.0
Gadidae	456	1.7	763	0.36
Macrouroididae	944	1.7	1,580	0.35
Macrouridae	661	1.7	1,107	31.1
Zoarcidae	480	1.7	803	0.39
Enoploteuthidae	19	2.31	44	1.1
Octopoteuthidae	89	3.08	274	0.14
Onychoteuthidae	123	4.53	555	0.26
Gonatidae	191	3.78	724	21.3
Histioteuthidae	152	2.65	402	0.24
Architeuthidae	14516	2.65	38,468	0.0081
Ommastrephidae	118	4.01	474	0.49
Chroteuthidae	38	2.31	89	0.098
Mastigoteuthidae	41	1.82	74	1.8
Cranchiidae	107	1.69	180	6.1
Vampyroteuthidae	267	2.31	617	0.065
Octopodidae	280	3.42	959	0.11
Alloposidae	552	2.88	1,589	0.041
Ocythoidae	6800	2.88	19,584	0.016

1420

Table S8. Prey size distribution and calorific values used for *B. bairdii* energetics modeling.

Family	W (g)	E (kJ/g)	E (kJ)	Percent of diet
Architeuthidae	24,000	2.65	63,600	0.33
Ommastrephidae	8,000	4.01	32,080	4.91
Lepidoteuthidae	2,000	4.01	8,020	0.64
Pholidoteuthidae	1,700	3.78	6,426	2.14
Vampyroteuthidae	1,000	2.31	2,310	0.44
Psychroteuthidae	1,000	2.65	2,650	0.03
Octopoteuthidae	1,000	3.08	3,080	21.02
Histioteuthidae	1,000	2.65	2,650	29.57
Ancistrocheiridae	800	2.31	1,848	9.40
Alloposidae	700	2.31	1,617	1.05
Onychoteuthidae	600	4.53	2,718	20.98
Cycloteuthidae	500	2.31	1,155	1.40
Octopodidae	500	3.77	1,885	0.00
Mastigoteuthidae	300	2.31	693	0.08
Gonatidae	200	3.78	756	0.33
Cranchiidae	200	1.69	338	4.91
Chroteuthidae	100	2.31	231	0.64
Unidentified	800	2.65	2,310	2.14

Table S9. Prey size distribution and calorific values used for *P. macrocephalus* energetics modeling.

1425

Genus species	Whales (n)	Recording duration (h)	Foraging dives (n)	Feeding events (n)	Tag types	Locations	Deployment details
<i>Phocoena phocoena</i>	8	165.4	3914	16115	DTAG	Denmark	Wisniewska et al. 2016
<i>Grampus griseus</i>	17	79	216	1518	DTAG	California ^a	Arranz et al. 2016, 2018, 2019
<i>Globicephala macrorhynchus</i>	2	9.6	5	21	DTAG	Azores	This study, see below
<i>Globicephala melas</i>	9	103.9	110	846	DTAG	Tarifa, Spain	This study, see below
<i>Ziphius cavirostris</i>	4	63.3	24	741	DTAG	California ^a ; Azores	DeRuiter et al. 2013 and this study, see details below
<i>Orcinus orca</i>	10	40.1	76	307	DTAG	Salish Sea	Holt et al. 2017
<i>Mesoplodon densirostris</i>	9	111.4	50	1302	DTAG	Canary Islands	Arranz et al. 2011
<i>Berardius bairdii</i>	1	14.6	2	70	DTAG	California ^a	Stimpert et al. 2014 and this study, see details below
<i>Physeter macrocephalus</i>	36	364.2	347	3853	DTAG	Dominica; Norway; California ^a	Tønnesen et al. 2018; Fais et al. 2015; Southall et al. 2012
<i>Eubalaena glacialis</i>	20	148.0	288	n/a	DTAG	NW Atlantic	Nowacek et al. 2001; Parks et al. 2011
<i>Balana mysticetus</i>	6	13.9	22	n/a	DTAG	Greenland	Simon et al. 2009
<i>Balaenoptera bonaerensis</i>	8	151.3	852	4726	CATS	Antarctica	This study, see details below
<i>Megaptera novaeangliae</i>	57	530.1	3162	9397	DTAG, CATS	Antarctica; Alaska; California; South Africa; Stellwagen Bank; Greenland	Burrows et al. 2016; Friedlaender et al. 2013; Cade et al. 2016
<i>Balaenoptera physalus</i>	23	296.7	771	3266	DTAG, CATS	California ^a ; Azores; Greenland; Stellwagen	Allen et al. 2016
<i>Balaenoptera musculus</i>	90	1016.7	3511	11208	DTAG, CATS	California ^a	Cade et al. 2016

^a Some tagged individuals were subjects in a behavioral response experiment (121), but we only used data before and after periods of sound exposure in the present study.

1430

Table S10. Foraging data set from suction-cup tagged cetaceans. Values indicate the total number of foraging dives and feeding events recorded from tag data. Details on tag settings, attachment and recovery methods can be found in the cited references (21, 33, 44, 85, 90, 92, 97, 102-104, 107, 114, 120, 121, 123, 124, 126, 182-185).

1435

group	fig.	model	MR exp.	Pagel's λ	intercept	slope	a	b	RSE	df
Rorquals	3	pGLS	n.a.	1	-3.3314 (-5.4463 - -1.7731)	1.8836 (1.5343 - 2.3601)	4.7e-04 (3.6e-06 - 0.0169)	1.8836	0.4702	13, 9
		OLS			-3.5089 (-5.6060 - -2.0407)	1.9293 (1.6031 - 2.3996)	3.1e-04 (2.5e-06 - 0.0091)	1.9293	0.4380	13, 9
Odontocetes		pGLS		1	-0.5252 (-1.4802 - 0.2083)	0.9811 (0.7891 - 1.2532)	0.2984 (0.0331 - 1.6153)	0.9811	0.4702	13, 9
		OLS			0.4487 (-0.0428 - 1.1061)	0.7395 (0.5556 - 0.8998)	2.8102 (0.9062 - 12.7687)	0.7395	0.4380	13, 9
Rorquals	4	pGLS	0.45	0.854	-0.8687 (-3.0660 - 0.7069)	0.6203 (0.2687 - 1.1057)	0.1353 (0.00086 - 5.0916)	0.6203	0.3616	13, 9
OLS		-1.1546 (-3.2920 - 0.3881)			0.6920 (0.3534 - 1.1684)	0.0700 (0.00051 - 2.4438)	0.6920	0.3008	13, 9	
Rorquals		pGLS	0.61	1	-0.9301 (-2.9702 - 0.7068)	0.6040 (0.2416 - 1.0608)	0.1175 (0.00107 - 5.0906)	0.6040	0.3707	13, 9
		OLS			-1.2869 (-3.3437 - 0.2135)	0.6936 (0.3615 - 1.1586)	0.0517 (0.00045 - 1.6349)	0.6936	0.3668	13, 9
Rorquals		pGLS	0.68	1	-1.0681 (-3.2324 - 0.5368)	0.6107 (0.2521 - 1.0927)	0.0855 (0.00059 - 3.4418)	0.6107	0.3223	13, 9
		OLS			-1.3654 (-3.5212 - 0.1441)	0.6861 (0.3560 - 1.1694)	0.0431 (3e-04 - 1.3935)	0.6861	0.3333	13, 9
Rorquals		pGLS	0.75	1	-1.1283 (-3.2486 - 0.5036)	0.5865 (0.2275 - 1.0699)	0.0744 (0.00056 - 3.1883)	0.5865	0.3271	13, 9
		OLS			-1.3744 (-3.4994 - 0.1281)	0.6493 (0.3169 - 1.1311)	0.0422 (0.00032 - 1.3430)	0.6493	0.3462	13, 9
Odontocetes		pGLS	0.45	0.854	1.1126 (0.4201 - 1.7249)	-0.1044 (-0.2707 - 0.1043)	12.9595 (2.6309 - 53.0772)	-	0.3616	13, 9
		OLS			1.3691 (0.8356 - 1.9225)	-0.1638 (-0.3202 - -0.0024)	23.3942 (6.8488 - 83.6489)	-	0.3008	13, 9
Odontocetes		pGLS	0.61	1	0.8771 (-0.0082 - 1.2758)	-0.1667 (-0.2801 - 0.0900)	7.5346 (0.9812 - 18.8693)	-	0.3707	13, 9
		OLS			1.4210 (0.9334 - 1.9962)	-0.3112 (-0.4743 - -0.1608)	26.3638 (8.5785 - 99.1244)	-	0.3668	13, 9
Odontocetes		pGLS	0.68	1	0.6565 (-0.2993 - 1.1853)	-0.1340 (-0.2740 - 0.1512)	4.5343 (0.5021 - 15.3209)	-	0.3223	13, 9
		OLS			1.1510 (0.6359 - 1.7456)	-0.2636 (-0.4300 - -0.1077)	14.1592 (4.3243 - 55.6615)	-	0.3333	13, 9
Odontocetes		pGLS	0.75	1	0.6051 (-0.2790 - 1.2063)	-0.1792 (-0.3362 - 0.0941)	4.0278 (0.5260 - 16.0791)	-	0.3271	13, 9
		OLS			1.1250 (0.6588 - 1.7316)	-0.3162 (-0.4868 - -0.1843)	13.3350 (4.5578 - 53.9016)	-	0.3462	13, 9

Table S11. Results of phylogenetic (pGLS) and ordinary (OLS) least squares analysis of cetacean foraging capacity. Figure 3: Scaling of prey energy (P_E) captured during each feeding event. Figure 4: Scaling of energetic efficiency for foraging dives. Regressions were determined for the energetic efficiency values (E_E) calculated among cetacean species for varying assumptions about how metabolic rate (MR) scales with body mass ($MR \propto Mc^{0.75}$; $MR \propto Mc^{0.68}$; $MR \propto Mc^{0.61}$; $MR \propto Mc^{0.45}$). Tabulated values are shown for the slope and intercept describing

regressions on log-transformed data space, as well as the transformation to allometric equations
1445 ($P_E = aM_c^b$ or $E_E = aM_c^b$). Numbers in brackets are 95% confidence intervals estimated from 10,000
bootstrap replicates using the biased-corrected and accelerated method. Values of Pagel's λ were
estimated using restricted maximum likelihood approach (under a Brownian model of evolution).
High values of λ indicate a strong similarity in the relationship between predictor and response
variables for closely related taxa. $\lambda = 0$ suggests that the predictor-response variable relationship
1450 is unrelated to phylogeny, while values of $\lambda > 1$ can arise if, for instance, traits are more similar
than predicted by Brownian motion, given the phylogeny. When the latter case occurred, λ was
fixed at 1. Further details are shown in R Markdown Supplement.

1455

1460

1465

1470

1475

1480

1485

1. G. J. Vermeij, Gigantism and its implications for the history of life. *PloS one* **11**, e0146092 (2016).
2. C. R. McClain *et al.*, Sizing ocean giants: patterns of intraspecific size variation in marine megafauna. *PeerJ* **3**, e715 (2015).
- 1490 3. C. Pimiento, J. L. Cantalapiedra, K. Shimada, D. J. Field, J. B. Smaers, Evolutionary pathways toward gigantism in sharks and rays. *Evolution* **73**, 588-599 (2019).
4. M. Friedman, Parallel evolutionary trajectories underlie the origin of giant suspension-feeding whales and bony fishes. *Proceedings of the Royal Society B-Biological Sciences* **279**, 944-951 (2011).
- 1495 5. W. Gearty, C. R. McClain, J. L. Payne, Energetic tradeoffs control the size distribution of aquatic mammals. *Proceedings of the National Academy of Sciences* **115**, 4191-4199 (2018).
6. M. Friedman *et al.*, 100-Million-Year Dynasty of Giant Planktivorous Bony Fishes in the Mesozoic Seas. *Science* **327**, 990-993 (2010).
- 1500 7. N. P. Kelley, N. D. Pyenson, Evolutionary innovation and ecology in marine tetrapods from the Triassic to the Anthropocene. *Science* **348**, aaa3716 (2015).
8. R. E. Fordyce, F. G. Marx, Gigantism precedes filter feeding in baleen whale evolution. *Curr. Biol.* **28**, 1670-1676. e1672 (2018).
- 1505 9. G. J. Slater, J. A. Goldbogen, N. D. Pyenson, Independent evolution of baleen whale gigantism linked to Plio-Pleistocene ocean dynamics. *Proc. R. Soc. B* **284**, 20170546 (2017).
10. O. Lambert *et al.*, The giant bite of a new raptorial sperm whale from the Miocene epoch of Peru. *Nature* **466**, 105 (2010).
- 1510 11. P. Domenici, The scaling of locomotor performance in predator-prey encounters: from fish to killer whales. *Comparative Biochemistry and Physiology a-Molecular and Integrative Physiology* **131**, 169-182 (2001).
12. F. H. Jensen, M. Johnson, M. Ladegaard, D. M. Wisniewska, P. T. Madsen, Narrow Acoustic Field of View Drives Frequency Scaling in Toothed Whale Biosonar. *Curr. Biol.* **28**, 3878-3885. e3873 (2018).
- 1515 13. J. Goldbogen, P. Madsen, The evolution of foraging capacity and gigantism in cetaceans. *Journal of Experimental Biology* **221**, jeb166033 (2018).
14. E. L. Hazen, A. S. Friedlaender, J. A. Goldbogen, Blue whales (*Balaenoptera musculus*) optimize foraging efficiency by balancing oxygen use and energy gain as a function of prey density. *Science Advances* **1**, e1500469 (2015).
- 1520 15. K. J. Benoit-Bird, B. L. Southall, M. A. Moline, Predator-guided sampling reveals biotic structure in the bathypelagic. *Proceedings of the Royal Society B-Biological Sciences* **283**, 20152457 (2016).
16. J. A. Goldbogen *et al.*, Scaling of lunge feeding performance in rorqual whales: mass-specific energy expenditure increases with body size and progressively limits diving capacity. *Functional Ecology* **26**, 216-226 (2012).
- 1525 17. .
18. M. Clarke, C. Lu, Vertical distribution of cephalopods at 18 N 25 W in the North Atlantic. *Journal of the Marine Biological Association of the United Kingdom* **55**, 165-182 (1975).
- 1530 19. M. R. Clarke, Cephalopods as prey. III. Cetaceans. *Phil. Trans. R. Soc. Lond. B* **351**, 1053-1065 (1996).

20. K. Aoki *et al.*, Active hunting by deep-diving sperm whales: 3D dive profiles and maneuvers during bursts of speed. *Marine Ecology Progress Series* **444**, 289-U322 (2012).
- 1535 21. A. Fais, M. Johnson, M. Wilson, N. Aguilar Soto, P. T. Madsen, Sperm whale predator-prey interactions involve chasing and buzzing, but no acoustic stunning. *Scientific Reports* **6**, 28562 (2016).
22. E. P. White, S. K. M. Ernest, A. J. Kerckhoff, B. J. Enquist, Relationships between body size and abundance in ecology. *Trends in Ecology & Evolution* **22**, 323-330 (2007).
- 1540 23. J. Potvin, A. J. Werth, Oral cavity hydrodynamics and drag production in Balaenid whale suspension feeding. *PloS one* **12**, e0175220 (2017).
24. K. J. Benoit-Bird, C. M. Waluk, J. P. Ryan, Forage Species Swarm in Response to Coastal Upwelling. *Geophysical Research Letters* **46**, 1537-1546 (2019).
25. R. B. Benson, G. Hunt, M. T. Carrano, N. Campione, Cope's rule and the adaptive landscape of dinosaur body size evolution. *Palaeontology* **61**, 13-48 (2018).
- 1545 26. P. M. Sander *et al.*, Biology of the sauropod dinosaurs: the evolution of gigantism. *Biological Reviews* **86**, 117-155 (2011).
27. G. P. Burness, J. Diamond, T. Flannery, Dinosaurs, dragons, and dwarfs: the evolution of maximal body size. *Proceedings of the National Academy of Sciences* **98**, 14518-14523 (2001).
- 1550 28. M. A. Tucker, T. L. Rogers, Examining predator-prey body size, trophic level and body mass across marine and terrestrial mammals. *Proceedings of the Royal Society B-Biological Sciences* **281**, 1-9 (2014).
29. C. K. Geijer, G. Notarbartolo di Sciara, S. Panigada, Mysticete migration revisited: are Mediterranean fin whales an anomaly? *Mammal Review* **46**, 284-296 (2016).
- 1555 30. E. Pirotta *et al.*, A Dynamic State Model of Migratory Behavior and Physiology to Assess the Consequences of Environmental Variation and Anthropogenic Disturbance on Marine Vertebrates. *The American Naturalist* **191**, E40-E56 (2018).
31. R. Williams *et al.*, Evidence for density-dependent changes in body condition and pregnancy rate of North Atlantic fin whales over four decades of varying environmental conditions. *ICES Journal of Marine Science* **70**, 1273-1280 (2013).
- 1560 32. J. H. Geisler, M. R. McGowen, G. Yang, J. Gatesy, A supermatrix analysis of genomic, morphological, and paleontological data from crown Cetacea. *BMC Evolutionary Biology* **11**, 112 (2011).
- 1565 33. Danuta M. Wisniewska *et al.*, Ultra-High Foraging Rates of Harbor Porpoises Make Them Vulnerable to Anthropogenic Disturbance. *Curr. Biol.* **26**, 1441-1446 (2016).
34. C. Wardle, Limit of fish swimming speed. *Nature* **255**, 725 (1975).
35. P. Börjesson, P. Berggren, B. Ganning, Diet of harbor porpoises in the Kattegat and Skagerrak seas: accounting for individual variation and sample size. *Marine Mammal Science* **19**, 38-058 (2003).
- 1570 36. B. M. Braune, Mercury accumulation in relation to size and age of Atlantic herring (*Clupea harengus harengus*) from the southwestern Bay of Fundy, Canada. *Archives of Environmental Contamination and Toxicology* **16**, 311-320 (1987).
37. F. Arrhenius, S. Hansson, Growth and seasonal changes in energy content of young Baltic Sea herring (*Clupea harengus* L.). *ICES Journal of Marine Science* **53**, 792-801 (1996).
- 1575

38. M. Würtz, R. Poggi, M. R. Clarke, Cephalopods from the stomachs of a Risso's dolphin (*Grampus griseus*) from the Mediterranean. *Journal of the Marine Biological Association of the United Kingdom* **72**, 861-867 (1992).
- 1580 39. M. Clarke, R. Young, Description and analysis of cephalopod beaks from stomachs of six species of odontocete cetaceans stranded on Hawaiian shores. *Journal of the Marine Biological Association of the United Kingdom* **78**, 623-641 (1998).
40. M. R. Clarke, A handbook for the identification of cephalopod beaks. (1986).
- 1585 41. H. Ohizumi, T. Isoda, T. Kishiro, H. Kato, Feeding habits of Baird's beaked whale *Berardius bairdii*, in the western North Pacific and Sea of Okhotsk off Japan. *Fisheries Science* **69**, 11-20 (2003).
42. M. Santos, V. Martin, M. Arbelo, A. Fernández, G. J. Pierce, Insights into the diet of beaked whales from the atypical mass stranding in the Canary Islands in September 2002. *Journal of the Marine Biological Association of the United Kingdom* **87**, 243-251 (2007).
- 1590 43. P. Madsen, M. Johnson, N. A. de Soto, W. Zimmer, P. Tyack, Biosonar performance of foraging beaked whales (*Mesoplodon densirostris*). *Journal of Experimental Biology* **208**, 181-194 (2005).
44. P. Arranz *et al.*, Following a foraging fish-finder: Diel habitat use of Blainville's beaked whales revealed by echolocation. *PloS one* **6**, e28353 (2011).
- 1595 45. J. Herman, A. Kitchener, J. Baker, C. Lockyer, The most northerly record of Blainville's beaked whale, *Mesoplodon densirostris*, from the eastern Atlantic. *Mammalia* **58**, 657-660 (1994).
46. D. P. Gannon, A. J. Read, J. E. Craddock, K. M. Fristrup, J. R. Nicolas, Feeding ecology of long-finned pilot whales *Globicephala melas* in the western North Atlantic. *Marine Ecology Progress Series* **148**, 1-10 (1997).
- 1600 47. D. P. Gannon, A. J. Ready, J. E. Craddock, J. G. Mead, Stomach contents of long-finned pilot whales (*Globicephala melas*) stranded on the US mid-Atlantic coast. *Marine Mammal Science* **13**, 405-418 (1997).
48. E. H. Sinclair, Stomach contents of four short-finned pilot whales (*Globicephala macrorhynchus*) from the southern California Bight. *Marine Mammal Science* **8**, 76-81 (1992).
- 1605 49. G. Desportes, R. Mouritsen, Diet of the pilot whale, *Globicephala melas*, around the Faroe Islands. *ICES CM*, 1988/N:1912 (1988).
- 1610 50. N. Aguilar de Soto *et al.*, Cheetahs of the deep sea: deep foraging sprints in short-finned pilot whales off Tenerife (Canary Islands). *Journal of Animal Ecology* **77**, 936-947 (2008).
51. V. Hernández-García, V. Martín, Stomach contents of two short-finned pilot whale (*Globicephala macrorhynchus* Gray, 1846) (Cetacea delphinidae) off the Canary Islands: A preliminary note. *International Council for the Exploration of the sea*. **16**, (1994).
- 1615 52. U. Piatkowski, V. Hernández-García, M. Clarke, On the biology of the European flying squid *Todarodes sagittatus* (Lamarck, 1798)(Cephalopoda, Ommastrephidae) in the central eastern Atlantic. *African Journal of Marine Science* **20**, 375-383 (1998).
- 1620 53. C. D. MacLeod, M. Santos, G. J. Pierce, Review of data on diets of beaked whales: evidence of niche separation and geographic segregation. *Journal of the Marine Biological Association of the United Kingdom* **83**, 651-665 (2003).

54. K. L. West, W. A. Walker, R. W. Baird, J. G. Mead, P. W. Collins, Diet of Cuvier's beaked whales *Ziphius cavirostris* from the North Pacific and a comparison with their diet world-wide. *Marine Ecology Progress Series* **574**, 227-242 (2017).
- 1625 55. M. B. Santos *et al.*, Feeding ecology of Cuvier's beaked whale (*Ziphius cavirostris*): a review with new information on the diet of this species. *Journal of the Marine Biological Association of the United Kingdom* **81**, 687-694 (2001).
56. J. K. B. Ford, B. M. Wright, G. M. Ellis, J. R. Candy, "Chinook salmon predation by resident killer whales: seasonal and regional selectivity, stock identity of prey, and consumption rates," (Canadian Science Advisory Secretariat, 2010).
- 1630 57. J. K. Ford, G. M. Ellis, Selective foraging by fish-eating killer whales *Orcinus orca* in British Columbia. *Marine Ecology Progress Series* **316**, 185-199 (2006).
58. S. M. O'Neill, G. M. Ylitalo, J. E. West, Energy content of Pacific salmon as prey of northern and southern resident killer whales. *Endangered Species Research* **25**, 265-281 (2014).
- 1635 59. M. M. Holt, M. B. Hanson, D. A. Giles, C. K. Emmons, J. T. Hogan, Noise levels received by endangered killer whales *Orcinus orca* before and after implementation of vessel regulations. *Endangered Species Research* **34**, 15-26 (2017).
60. M. J. Ford *et al.*, Estimation of a killer whale (*Orcinus orca*) population's diet using sequencing analysis of DNA from feces. *Plos one* **11**, e0144956 (2016).
- 1640 61. M. B. Hanson *et al.*, Species and stock identification of prey consumed by endangered southern resident killer whales in their summer range. *Endangered Species Research* **11**, 69-82 (2010).
62. M. Clarke, C. Roper, Cephalopods represented by beaks in the stomach of a sperm whale stranded at Paekakariki, North Island, New Zealand. *African Journal of Marine Science* **20**, (1998).
- 1645 63. H. Whitehead, *Sperm whales: social evolution in the ocean*. (University of Chicago press, 2003).
64. A. Clarke, M. Clarke, L. J. Holmes, T. Waters, Calorific values and elemental analysis of eleven species of oceanic squids (Mollusca: Cephalopoda). *Journal of the Marine Biological Association of the United Kingdom* **65**, 983-986 (1985).
- 1650 65. D. Demer *et al.*, Calibration of acoustic instruments. *ICES Cooperative Research Report* **326**, 133 (2015).
66. T. Jarvis, N. Kelly, S. Kawaguchi, E. van Wijk, S. Nicol, Acoustic characterisation of the broad-scale distribution and abundance of Antarctic krill (*Euphausia superba*) off East Antarctica (30-80 E) in January-March 2006. *Deep Sea Research Part II: Topical Studies in Oceanography* **57**, 916-933 (2010).
- 1655 67. J. Potvin, J. A. Goldbogen, R. E. Shadwick, Scaling of lunge feeding in rorqual whales: An integrated model of engulfment duration. *Journal of Theoretical Biology* **267**, 437-453 (2010).
- 1660 68. J. Potvin, J. A. Goldbogen, R. E. Shadwick, Metabolic Expenditures of Lunge Feeding Rorquals Across Scale: Implications for the Evolution of Filter Feeding and the Limits to Maximum Body Size. *PLoS ONE* **7**, e44854 (2012).
69. L. McGarry, An Examination Of Blue Whale Foraging And Its Krill Prey Field In The Monterey Bay Submarine Canyon. (2014).

- 1665 70. T. K. Stanton, D. Chu, P. H. Wiebe, Sound scattering by several zooplankton groups. II. Scattering models. *The Journal of the Acoustical Society of America* **103**, 236-253 (1998).
71. J. A. Goldbogen *et al.*, Mechanics, hydrodynamics and energetics of blue whale lunge feeding: efficiency dependence on krill density. *Journal of Experimental Biology* **214**, 131-146 (2011).
- 1670 72. T. Nemoto, Net sampling and abundance assessment of euphausiids. *Biological Oceanography* **2**, 211-226 (1983).
73. J. Färber-Lorda, R. Gaudy, P. Mayzaud, Elemental composition, biochemical composition and caloric value of Antarctic krill.: Implications in Energetics and carbon balances. *Journal of Marine Systems* **78**, 518-524 (2009).
- 1675 74. A. Clarke, Lipid content and composition of Antarctic krill, *Euphausia superba* Dana. *Journal of Crustacean Biology* **4**, 285-294 (1984).
75. A. Clarke, The biochemical composition of krill, *Euphausia superba* Dana, from South Georgia. *Journal of Experimental Marine Biology and Ecology* **43**, 221-236 (1980).
- 1680 76. E. M. Chenoweth, University of Alaska Fairbanks, (2018).
77. M. F. Baumgartner, B. R. Mate, Summertime foraging ecology of North Atlantic right whales. *Marine Ecology Progress Series* **264**, 123-135 (2003).
78. C. A. Mayo, M. K. Marx, Surface foraging behaviour of the North Atlantic right whale, *Eubalaena glacialis*, and associated zooplankton characteristics. *Canadian Journal of Zoology* **68**, 2214-2220 (1990).
- 1685 79. K. L. Laidre, M. P. Heide-Jørgensen, T. G. Nielsen, Role of the bowhead whale as a predator in West Greenland. *Marine Ecology Progress Series* **346**, 285-297 (2007).
80. N. J. Karnovsky, S. Kwaśniewski, J. M. Węśławski, W. Walkusz, A. Beszczyńska-Möller, Foraging behavior of little auks in a heterogeneous environment. *Marine Ecology Progress Series* **253**, 289-303 (2003).
- 1690 81. G. Comita, S. Marshall, A. Orr, On the biology of *Calanus finmarchicus*: XIII. Seasonal change in weight, calorific value and organic matter. *Journal of the Marine Biological Association of the United Kingdom* **46**, 1-17 (1966).
82. M. Johnson, P. L. Tyack, A digital acoustic recording tag for measuring the response of wild marine mammals to sound. *IEEE Journal of Oceanic Engineering* **28**, 3-12 (2003).
- 1695 83. M. Simon, M. Johnson, P. T. Madsen, Keeping momentum with a mouthful of water: Behavior and kinematics of humpback whale lunge feeding. *Journal of Experimental Biology* **215**, 3786-3798 (2012).
84. K. S. Ydesen *et al.*, What a jerk: prey engulfment revealed by high-rate, super-cranial accelerometry on a harbour seal (*Phoca vitulina*). *Journal of Experimental Biology* **217**, 2239-2243 (2014).
- 1700 85. P. Arranz *et al.*, Discrimination of fast click series produced by tagged Risso's dolphins (*Grampus griseus*) for echolocation or communication. *Journal of Experimental Biology*, jeb. 144295 (2016).
- 1705 86. P. M. Sørensen *et al.*, Click communication in wild harbour porpoises (*Phocoena phocoena*). *Scientific Reports* **8**, 9702 (2018).
87. J. Croxall, P. Prince, California content of squid (Mollusca: Cephalopoda). *BAS Bulletins* **55**, 27-31 (1982).
- 1710 88. G. Sanchez *et al.*, Genus-level phylogeny of cephalopods using molecular markers: current status and problematic areas. *PeerJ* **6**, e4331 (2018).

89. J. J. Childress, M. H. Nygaard, The chemical composition of midwater fishes as a function of depth of occurrence off southern California. *Deep Sea Research and Oceanographic Abstracts* **20**, 1093-1109 (1973).
- 1715 90. A. K. Stimpert *et al.*, Acoustic and foraging behavior of a Baird's beaked whale, *Berardius bairdii*, exposed to simulated sonar. *Scientific Reports* **4**, 7031 (2014).
91. A. S. Friedlaender *et al.*, Feeding rates and under-ice foraging strategies of the smallest lunge filter feeder, the Antarctic minke whale (*Balaenoptera bonaerensis*). *Journal of Experimental Biology* **217**, 2851-2854 (2014).
- 1720 92. D. E. Cade, A. S. Friedlaender, J. Calambokidis, J. A. Goldbogen, Kinematic Diversity in Rorqual Whale Feeding Mechanisms. *Curr. Biol.* **26**, 2617-2624 (2016).
93. J. Goldbogen *et al.*, Using Digital Tags With Integrated Video and Inertial Sensors to Study Moving Morphology and Associated Function in Large Aquatic Vertebrates. *The Anatomical Record* **300**, 1935-1941 (2017).
- 1725 94. J. A. Goldbogen *et al.*, Kinematics of foraging dives and lunge-feeding in fin whales. *Journal of Experimental Biology* **209**, 1231-1244 (2006).
95. J. A. Goldbogen *et al.*, Underwater acrobatics by the world's largest predator: 360 degrees rolling manoeuvres by lunge-feeding blue whales. *Biology Letters* **9**, 20120986 (2013).
- 1730 96. J. Goldbogen *et al.*, How baleen whales feed: the biomechanics of engulfment and filtration. *Annual Review of Marine Science* **9**, 367-386 (2017).
97. A. N. Allen, J. A. Goldbogen, A. S. Friedlaender, J. Calambokidis, Development of an automated method of detecting stereotyped feeding events in multisensor data from tagged rorqual whales. *Ecology and Evolution* **6**, 7522-7535 (2016).
- 1735 98. A. S. Friedlaender, J. A. Goldbogen, E. L. Hazen, J. Calambokidis, B. L. Southall, Feeding performance by sympatric blue and fin whales exploiting a common prey resource. *Marine Mammal Science* **31**, 345-354 (2015).
99. A. S. Friedlaender *et al.*, Prey-mediated behavioral responses of feeding blue whales in controlled sound exposure experiments. *Ecological Applications*, (2016).
- 1740 100. J. A. Goldbogen *et al.*, Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *BioScience* **63**, 90-100 (2013).
101. J. A. Goldbogen *et al.*, Prey density and distribution drive the three-dimensional foraging strategies of the largest filter feeder. *Functional Ecology* **29**, 951-961 (2015).
- 1745 102. A. S. Friedlaender, R. B. Tyson, A. K. Stimpert, A. J. Read, D. P. Nowacek, Extreme diel variation in the feeding behavior of humpback whales along the western Antarctic Peninsula during autumn. *Marine Ecology Progress Series* **494**, 281-289 (2013).
103. P. Arranz *et al.*, Diving behavior and fine-scale kinematics of free-ranging Risso's dolphins foraging in shallow and deep-water habitats. **7**, 53 (2019).
- 1750 104. P. Arranz *et al.*, Risso's dolphins plan foraging dives. *Journal of Experimental Biology* **221**, jeb165209 (2018).
105. D. M. Wisniewska *et al.*, Response to "Resilience of harbor porpoises to anthropogenic disturbance: Must they really feed continuously?". *Marine Mammal Science* **34**, 265-270 (2018).
- 1755 106. J. B. Tennessen *et al.*, Kinematic signatures of prey capture from archival tags reveal sex differences in killer whale foraging activity. *The Journal of Experimental Biology* **222**, jeb191874 (2019).

107. M. M. Holt, M. B. Hanson, D. A. Giles, C. K. Emmons, J. T. J. E. S. R. Hogan, Noise levels received by endangered killer whales *Orcinus orca* before and after implementation of vessel regulations. **34**, 15-26 (2017).
- 1760 108. J. Houghton *et al.*, The Relationship between Vessel Traffic and Noise Levels Received by Killer Whales (*Orcinus orca*). *PLOS ONE* **10**, e0140119 (2015).
109. M. Johnson, N. Aguilar de Soto, P. T. Madsen, Studying the behaviour and sensory ecology of marine mammals using acoustic recording tags: a review. *Marine Ecology-Progress Series* **395**, 55-73 (2009).
- 1765 110. P. Verborgh *et al.*, in *Advances in Marine Biology*, G. Notarbartolo Di Sciara, M. Podestà, B. E. Curry, Eds. (Academic Press, 2016), vol. 75, pp. 173-203.
111. P. Verborgh *et al.*, Survival rate, abundance, and residency of long-finned pilot whales in the Strait of Gibraltar. *Marine Mammal Science* **25**, 523-536 (2009).
112. R. de Stephanis *et al.*, Long-term social structure of long-finned pilot whales (*Globicephala melas*) in the Strait of Gibraltar. *Acta Ethologica* **11**, 81 (2008).
- 1770 113. F. Visser *et al.*, The social context of individual foraging behaviour in long-finned pilot whales (*Globicephala melas*). *Behaviour* **151**, 1453-1477 (2014).
114. S. L. DeRuiter *et al.*, First direct measurements of behavioural responses by Cuvier's beaked whales to mid-frequency active sonar. *Biology Letters* **9**, (2013).
- 1775 115. F. Visser, "Off-range beaked whale study: Behavior and demography of Cuvier's beaked whale at the Azores (North Atlantic)," (Annual Report to Office of Naval Research (ONR), 2018).
116. S. Minamikawa, T. Iwasaki, T. Kishiro, Diving behaviour of a Baird's beaked whale, *Berardius bairdii*, in the slope water region of the western North Pacific: first dive records using a data logger. *Fisheries Oceanography* **16**, 573-577 (2007).
- 1780 117. S. Gero *et al.*, Behavior and social structure of the sperm whales of Dominica, West Indies. **30**, 905-922 (2014).
118. A. S. Friedlaender *et al.*, Diel changes in humpback whale *Megaptera novaeangliae* feeding behavior in response to sand lance *Ammodytes* spp. behavior and distribution. *Marine Ecology-Progress Series* **395**, 91-100 (2009).
- 1785 119. C. Ware, A. S. Friedlaender, D. P. Nowacek, Shallow and deep lunge feeding of humpback whales in fjords of the West Antarctic Peninsula. *Marine Mammal Science* **27**, 587-605 (2011).
120. J. Burrows *et al.*, Prey density and depth affect the fine-scale foraging behavior of humpback whales *Megaptera novaeangliae* in Sitka Sound, Alaska, USA. *Marine Ecology Progress Series* **561**, 245-260 (2016).
- 1790 121. B. L. Southall, D. Moretti, B. Abraham, J. Calambokidis, P. Tyack, Marine mammal behavioral response studies in Southern California: advances in technology and experimental methods. *Marine Technology Society Journal* **46**, 46-59 (2012).
122. J. A. Goldbogen *et al.*, Using accelerometers to determine the calling behavior of tagged baleen whales. *The Journal of Experimental Biology* **217**, 2449-2455 (2014).
- 1795 123. J. A. Goldbogen *et al.*, Blue whales respond to simulated mid-frequency military sonar. *Proceedings of the Royal Society B-Biological Sciences* **280**, (2013).
124. M. Simon, M. Johnson, P. Tyack, P. T. Madsen, Behaviour and kinematics of continuous ram filtration in bowhead whales (*Balaena mysticetus*). *Proceedings of the Royal Society B-Biological Sciences* **276**, 3819-3828 (2009).
- 1800

125. A. E. Nousek-McGregor, C. A. Miller, M. J. Moore, D. P. Nowacek, Effects of Body Condition on Buoyancy in Endangered North Atlantic Right Whales*. *Physiological and Biochemical Zoology* **87**, 160-171 (2014).
- 1805 126. D. P. Nowacek *et al.*, Buoyant balaenids: the ups and downs of buoyancy in right whales. *Proceedings of the Royal Society B-Biological Sciences* **268**, 1811-1816 (2001).
127. A. J. Werth, J. Potvin, Baleen Hydrodynamics and Morphology of Cross-Flow Filtration in Balaenid Whale Suspension Feeding. *PLOS ONE* **11**, e0150106 (2016).
128. A. J. Werth, Models of hydrodynamic flow in the bowhead whale filter feeding apparatus. *Journal of Experimental Biology* **207**, 3569-3580 (2004).
- 1810 129. J. M. van der Hoop *et al.*, Foraging rates of ram-filtering North Atlantic right whales. *Functional Ecology* **33**, 1290-1306 (2019).
130. J. Potvin, J. A. Goldbogen, R. E. Shadwick, Passive versus active engulfment: Verdict from trajectory simulations of lunge-feeding fin whales *Balaenoptera physalus*. *Journal of the Royal Society Interface* **6**, 1005-1025 (2009).
- 1815 131. J. Potvin, J. A. Goldbogen, R. E. Shadwick, in *The 20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*. (Seattle, WA, 2009).
132. C. D. Marshall, J. A. Goldbogen, in *Marine Mammal Physiology: Requisites for Ocean Living*, M. A. Castellini, J. Mellish, Eds. (CRC Press, Boca Raton, FL, 2015), chap. 5, pp. 95-118.
- 1820 133. A. C. Gleiss, J. Potvin, J. A. Goldbogen, Physical trade-offs shape the evolution of buoyancy control in sharks. *Proceedings of the Royal Society B-Biological Sciences* **284**, 20171345 (2017).
134. T. M. Williams, J. L. Maresh, Exercise energetics. *Marine Mammal Physiology. Requisites for Ocean Living*, 47-68 (2015).
- 1825 135. R. O'dor, D. Webber, The constraints on cephalopods: why squid aren't fish. *Canadian Journal of Zoology* **64**, 1591-1605 (1986).
136. W. A. Walker, J. G. Mead, R. L. Brownell Jr, Diets of Baird's beaked whales, *Berardius bairdii*, in the southern Sea of Okhotsk and off the Pacific coast of Honshu, Japan. *Marine Mammal Science* **18**, 902-919 (2002).
- 1830 137. P. Domenici, N. Herbert, C. Lefrançois, J. F. Steffensen, D. McKenzie, in *Swimming physiology of fish*. (Springer, 2013), pp. 129-159.
138. Y. Y. Watanabe *et al.*, Scaling of swim speed in breath-hold divers. *Journal of Animal Ecology* **80**, 57-68 (2011).
- 1835 139. N. D. Pyenson, S. N. Sponberg, Reconstructing Body Size in Extinct Crown Cetacea (Neoceti) Using Allometry, Phylogenetic Methods and Tests from the Fossil Record. *Journal of Mammalian Evolution* **18**, 269-288 (2011).
140. P. Domenici, The Visually Mediated Escape Response in Fish: Predicting Prey Responsiveness and the Locomotor Behaviour of Predators and Prey. *Marine and Freshwater Behaviour and Physiology* **35**, 87-110 (2002).
- 1840 141. A. P. French, *Newtonian Mechanics*. MIT Introductory Physics Series (WW Norton & Company Inc New York, 1971).
142. S. F. Hoerner, *Fluid Dynamic Drag*. (Published by the author., 1965).
143. R. D. Blevins, *Applied fluid dynamics handbook*. New York, Van Nostrand Reinhold Co., 1984, 568 p. **1**, (1984).

- 1845 144. J. A. Goldbogen, F. E. Fish, J. Potvin, in *Marine Mammal Physiology: Requisites for Ocean Living*, M. A. Castellini, J. Mellish, Eds. (CRC Press, Boca Raton, FL, 2015), chap. 1, pp. 3-28.
145. M. J. Lighthill, Note on the swimming of slender fish. *Journal of Fluid Mechanics* **9**, 305-317 (1960).
- 1850 146. F. E. Fish, J. J. Rohr, "Review of dolphin hydrodynamics and swimming performance," *SPAWARS System Center Technical Report* (San Diego, CA, 1999).
147. F. E. Fish, Comparative kinematics and hydrodynamics of odontocete cetaceans: Morphological and ecological correlates with swimming performance. *Journal of Experimental Biology* **201**, 2867-2877 (1998).
- 1855 148. F. E. Fish, Power output and propulsive efficiency of swimming bottlenose dolphins (*Tursiops truncatus*). *Journal of Experimental Biology* **185**, 179-193 (1993).
149. H. Hertel, *Structure, Form, Movement*. (Reinhold, New York, 1966).
150. F. E. Fish, Influence of hydrodynamic design and propulsive mode on mammalian swimming energetics. *Australian Journal of Zoology* **42**, 79-101 (1993).
- 1860 151. B. K. Ahlborn, R. W. Blake, K. H. S. Chan, Optimal fineness ratio for minimum drag in large whales. *Canadian Journal of Zoology-Revue Canadienne De Zoologie* **87**, 124-131 (2009).
152. T. Sarpkaya, *Wave forces on offshore structures*. (Cambridge University Press, 2010).
- 1865 153. J. L. Maresh *et al.*, Free-swimming northern elephant seals have low field metabolic rates that are sensitive to an increased cost of transport. *The Journal of Experimental Biology* **217**, 1485-1495 (2014).
154. A. J. Werth *et al.*, Filtration area scaling and evolution in mysticetes: trophic niche partitioning and the curious cases of sei and pygmy right whales. *Biological Journal of the Linnean Society* **125**, 264-279 (2018).
- 1870 155. C. H. Lockyer, Growth and Energy Budgets of Large Baleen Whales from the Southern Hemisphere. *Mammals in the Seas* **3**, 379-487 (1981).
156. P. W. Webb, Hydrodynamics and energetics of fish propulsion. *Bull. Fish. Res. Bd. Can.* **190**, 1-158 (1975).
- 1875 157. D. Weihs, Energetic advantages of burst swimming of fish. *Journal of Theoretical Biology* **48**, 215-229 (1974).
158. M. Kleiber, *The Fire of Life: an Introduction to Animal Energetics*. (R.E. Kteiger Publishing Co., Huntington, NY, 1975).
159. A. M. Hemmingsen, Energy metabolism as related to body size and respiratory surface, and its evolution. *Reports of the Steno Memorial Hospital (Copenhagen)* **13**, 1-110 (1960).
- 1880 160. J. M. van der Hoop *et al.*, Drag from fishing gear entangling North Atlantic right whales. **32**, 619-642 (2016).
161. A. E. Nousek-McGregor, Duke University, (2010).
- 1885 162. G. Ribak, D. Weihs, Z. J. J. o. E. B. Arad, Submerged swimming of the great cormorant *Phalacrocorax carbo sinensis* is a variant of the burst-and-glide gait. **208**, 3835-3849 (2005).
163. B. L. Woodward, J. P. Winn, F. E. Fish, Morphological specializations of baleen whales associated with hydrodynamic performance and ecological niche. *Journal of Morphology* **267**, 1284-1294 (2006).

- 1890 164. C. Lockyer, Body Weights of Some Species of Large Whales. *ICES Journal of Marine Science* **36**, 259-273 (1976).
165. L. S. Orton, P. F. Brodie, Engulfing Mechanics of Fin Whales. *Canadian Journal of Zoology* **65**, 2898-2907 (1987).
- 1895 166. R. E. Shadwick, J. A. Goldbogen, J. Potvin, N. D. Pyenson, A. W. Vogl, Novel muscle and connective tissue design enables high extensibility and controls engulfment volume in lunge-feeding rorqual whales. *The Journal of Experimental Biology* **216**, 2691-2701 (2013).
167. J. A. Goldbogen, N. D. Pyenson, R. E. Shadwick, Big gulps require high drag for fin whale lunge feeding. *Marine Ecology-Progress Series* **349**, 289-301 (2007).
- 1900 168. J. A. Goldbogen, J. Potvin, R. E. Shadwick, Skull and buccal cavity allometry increase mass-specific engulfment capacity in fin whales. *Proceedings of the Royal Society B-Biological Sciences* **277**, 861-868 (2010).
169. P. Scholander, Experimental investigations on the respiratory function in diving mammals and birds. *Hvalrådets skrifter* **22**, 1-131 (1940).
- 1905 170. B. Efron, Better bootstrap confidence intervals. *Journal of the American statistical Association* **82**, 171-185 (1987).
171. J. Felsenstein, Phylogenies and the comparative method. *American Naturalist* **125**, 1-15 (1985).
- 1910 172. E. Paradis, *Analysis of phylogenetics and evolution with R*. Use R! (Springer Science and Business Media, New York, NY, 2011), pp. 386.
173. J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, R Development Core Team. (R package version 3.1-104, 2012).
174. M. R. McGowen, M. Spaulding, J. Gatesy, Divergence date estimation and a comprehensive molecular tree of extant cetaceans. *Molecular Phylogenetics and Evolution* **53**, 891-906 (2009).
- 1915 175. W. H. Piel *et al.*, TreeBASE v. 2: a database of phylogenetic knowledge. (2009).
176. W. P. Maddison, D. R. Maddison. (2018).
177. L. Revell, Phylogenetic signal and linear regression on species data. *Methods in Ecology and Evolution* **1**, 319-329 (2010).
- 1920 178. E. Paradis, K. Schliep, ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* **35**, 526-528 (2018).
179. R. P. Freckleton, P. H. Harvey, M. Pagel, Phylogenetic Analysis and Comparative Data: A Test and Review of Evidence. *The American Naturalist* **160**, 712-726 (2002).
- 1925 180. T. Garland, A. R. Ives, Using the past to predict the present: Confidence intervals for regression equations in phylogenetic comparative methods. *American Naturalist* **155**, 346-364 (2000).
181. S. Wood, M. S. Wood, Package ‘mgcv’. *R package version* **1**, 29 (2015).
182. P. L. Tyack *et al.*, Beaked Whales Respond to Simulated and Actual Navy Sonar. *Plos One* **6**, e17009 (2011).
- 1930 183. A. Fais *et al.*, Sperm whale echolocation behaviour reveals a directed, prior-based search strategy informed by prey distribution. *Behavioral Ecology and Sociobiology* **69**, 663-674 (2015).
184. S. E. Parks, J. D. Warren, S. K., C. A. Mayo, D. Wiley, Dangerous dining: surface foraging of North Atlantic right whales increases risk of vessel collisions. *Biology Letters*. 2011 (10.1098/rsbl.2011.0578).
- 1935

185. P. Tønnesen, S. Gero, M. Ladegaard, M. Johnson, P. T. Madsen, First-year sperm whale calves echolocate and perform long, deep dives. *Behavioral ecology and sociobiology* **72**, 165 (2018).

pGLS for scaling of foraging capacity in cetaceans - **Figure 3** prey energy vs body mass

Sep 20, 2019

Contents

1	Background	2
2	Load libraries	2
2.1	References	2
3	Read in tree data	3
4	Read in and explore trait data	3
4.1	Read in data	3
5	Compute (weighted) mean values and store them in a new data frame	3
5.1	Rearrange the row order in df.spec to match mytree	4
5.2	Get rid of rows with NAs - subset the data	4
5.3	Adjust tree - drop species for which data are missing	4
5.4	Rearrange the row order in smydata to match smytree	5
5.5	Plot the data	5
6	Run OLS with feeding mode as a categorical predictor	7
6.1	Run OLS and model reduction using ML	7
6.2	Estimate final model using REML	8
6.3	Evaluate for phylogenetic correlation	10
7	Run pGLS with feeding mode as a categorical predictor	11
7.1	Estimate Pagel's λ (amount of phylogenetic signal) for each trait separately	11
7.2	Plot likelihood surface for Pagel's λ for model without feeding mode as a covariate	11
7.3	Estimate Pagel's λ for full model using REML	12
7.4	Plot likelihood surface for Pagel's λ - our estimate marked in red	13
7.5	Run pGLS and model reduction with a fixed Pagel's λ (using ML)	14
7.6	Estimate final model using REML	17
8	Estimate confidence intervals by bootstrapping	18
8.1	Bootstrap and compute percentile confidence intervals	18
8.2	Compute BCa (bias-corrected and accelerated) confidence intervals	21
8.3	Plot OLS model	26
8.4	Plot pGLS model	29
9	Extract summary statistics	32
10	Plot best models (OLS - dashed, PGLS - solid)	35
10.1	Construct output table	36

1 Background

This is an R Markdown file documenting the PGLS analysis of scaling relationships in dive and foraging performance data from the two groups of cetaceans: Odontocetes and Mysticetes (Rorquals).

2 Load libraries

```
library(AICcmodavg) # Mazerolle 2017
library(ape) # Paradis and Schliep 2018
library(nlme) # Pinheiro et al. 2018
library(phytools) # Revell 2012
library(geiger) # Harmon et al. 2008
library(dplyr) # Wickham et al. 2018
library(ggplot2) # Wickham 2016
library(lme4) # Bates et al. 2015
library(rptR) # Stoffel et al. 2017
library(knitr) # Xie 2014, 2015, 2018
library(car) # Fox and Weisberg 2011
library(tinytex) # Xie 2019
library(kableExtra) # Zhu 2019
```

2.1 References

- Bates, D., Maechler, M., Bolker, B. and Walker, S. (2015) Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1), 1-48. doi:10.18637/jss.v067.i01.
- Fox, J. and Weisberg, S. (2011) *An {R} Companion to Applied Regression*, Second Edition. Thousand Oaks CA: Sage. URL: <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion>
- Harmon, L.J., Weir, J.T., Brock, C.D., Glor, R.E. and Challenger, W. (2008) GEIGER: investigating evolutionary radiations. *Bioinformatics* 24:129-131.
- Mazerolle, M.J. (2017) AICcmodavg: Model selection and multimodel inference based on (Q)AIC(c). R package version 2.1-1. <https://cran.r-project.org/package=AICcmodavg>.
- Paradis, E. and Schliep, K. (2018) ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* 35: 526-528.
- Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D. and R Core Team (2018) nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-137, <URL: <https://CRAN.R-project.org/package=nlme>>.
- Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.* 3 217-223. doi:10.1111/j.2041-210X.2011.00169.x
- Stoffel, M. A., Nakagawa, S. and Schielzeth, H. (2017) rptR: repeatability estimation and variance decomposition by generalized linear mixed-effects models. *Methods Ecol Evol*, 8: 1639-1644. doi:10.1111/2041-210X.12797
- Wickham, H. (2016) ggplot2: Elegant graphics for data analysis. Springer-Verlag New York
- Wickham, H., Romain, F., Lionel, H. and Müller, K. (2018) dplyr: A grammar of data manipulation. R package version 0.7.8. <https://CRAN.R-project.org/package=dplyr>
- Xie, Y. (2014) knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595
- Xie, Y. (2015) *Dynamic documents with R and knitr*. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963
- Xie, Y. (2018) knitr: A general-purpose package for dynamic report generation in R. R package version 1.21.
- Xie, Y. (2019) tinytex: Helper Functions to Install and Maintain ‘TeX Live’, and Compile ‘LaTeX’ Documents. R package version 0.10. <https://CRAN.R-project.org/package=tinytex>
- Zhu, H. (2019) kableExtra: Construct Complex Table with ‘kable’ and Pipe Syntax. R package version 1.0.1. <https://CRAN.R-project.org/package=kableExtra>

3 Read in tree data

We used phylogenetic data produced by McGowen et al. (2009), downloaded as a Nexus file from TreeBase. The tree was edited in Mesquite to remove species without data on foraging rates, and to add timing of diversification of branches based on data published in McGowen et al. (2009).

```
mytree <- read.nexus("S10190_foragingsubset_Nov2018_edited.nex")
# plot(mytree)
```

4 Read in and explore trait data

4.1 Read in data

```
# d_full <- read.csv("Cetacea model output BOUT_EXTANT.csv")
d_full <- read.csv("Cetacea model output v10.13 Zc fix_BOUT_EXTANT.csv")

sub <- unique(d_full$MR.exponent)
d_full <- d_full[d_full$MR.exponent==sub[1],]

d_full$MR.exponent <- as.factor(d_full$MR.exponent)
d_full$M..kg. <- as.numeric(d_full$M..kg.)
d_full$Prey.W..g. <- as.numeric(d_full$Prey.W..g.)
d_full$Group <- ifelse(d_full$Family == "Balaenopteridae", "Rorqual",
                      ifelse(d_full$Family == "Balaenidae", "Balaenid", "Odontocete"))
d_full$Spec <- paste(d_full$Genus, d_full$Species, sep="_")

abb <- character(nrow(d_full))
for (i in seq(1,nrow(d_full))) {
  if (d_full$Genus[i] == "Globicephala" || d_full$Genus[i] == "Berardius" ||
      d_full$Species[i] == "bonaerensis") {
    abb[i] <- paste(substr(d_full$Genus[i],1,1), substr(d_full$Species[i],1,2), sep = "")
  } else {
    abb[i] <- paste(substr(d_full$Genus[i],1,1), substr(d_full$Species[i],1,1), sep = "")
  }
}

d_full$abbreviation <- abb

d_full$x <- log10(d_full$M..kg.)
d_full$y <- log10(d_full$Energy..kJ.)
spec <- unique(d_full$Spec)
```

5 Compute (weighted) mean values and store them in a new data frame

Weights based on relative frequency of occurrence estimated from stomach content data and prey mapping data for odontocetes and mysticetes.

```
x_mean <- tapply(d_full$x, d_full$Spec, mean)
y_mean <- by(d_full, d_full$Spec, with, weighted.mean(y, Percent))

gr <- tapply(d_full$Family, d_full$Spec, unique)
tx <- tapply(d_full$Group, d_full$Spec, unique)
```

```

fm <- tx
fm[tx=="Rorqual"] <- "Filter"
fm[tx=="Odontocete"] <- "Single-prey"
abbreviation <- tapply(d_full$abbreviation, d_full$Spec, unique)

data.spec <- cbind(gr,x_mean,y_mean)
df.spec <- data.frame(species=row.names(data.spec),data.spec,
                      row.names = rownames(data.spec),
                      check.rows = TRUE, check.names = TRUE)

df.spec$gr <- factor(df.spec$gr)
df.spec$fm <- factor(fm)
df.spec$abbreviation <- factor(abbreviation)
df.spec$Group <- tx

```

5.1 Rearrange the row order in df.spec to match mytree

```
df.spec <- df.spec[match(mytree$tip.label,rownames(df.spec)),]
```

5.2 Get rid of rows with NAs - subset the data

```

smydata <- df.spec
smydata <- smydata[!is.na(smydata$y_mean),]
smydata <- smydata[!is.na(smydata$x_mean),]
smydata$fm <- factor(smydata$fm)
smydata$Group <- smydata$Group

```

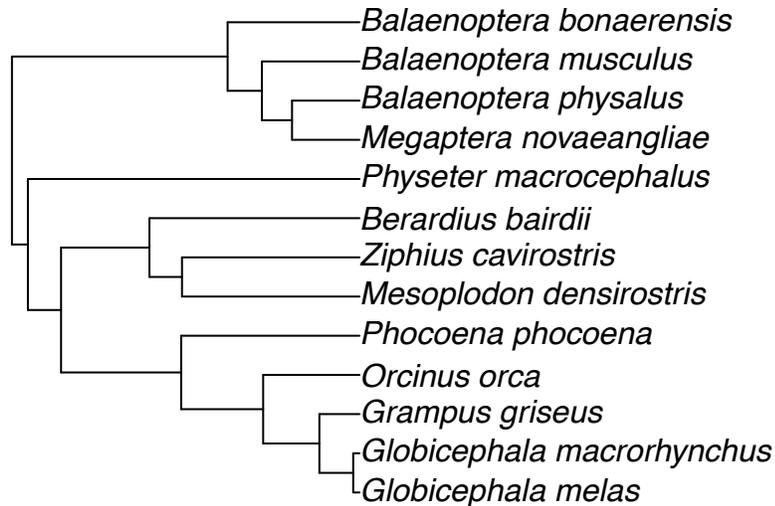
5.3 Adjust tree - drop species for which data are missing

```

pruned.tree <- drop.tip(mytree,mytree$tip.label[!match(smydata$species,
                                                       mytree$tip.label)])

smytree <- pruned.tree
plot(smytree)

```



5.4 Rearrange the row order in smydata to match smytree

```
smydata <- smydata[match(smytree$tip.label,rownames(smydata)),]
rownames(smydata) <- NULL
kable(smydata,
      caption = "Model inputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")
```

```
rownames(smydata) <- smydata$species
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure3_smydata.rds")
saveRDS(smydata,fil)
```

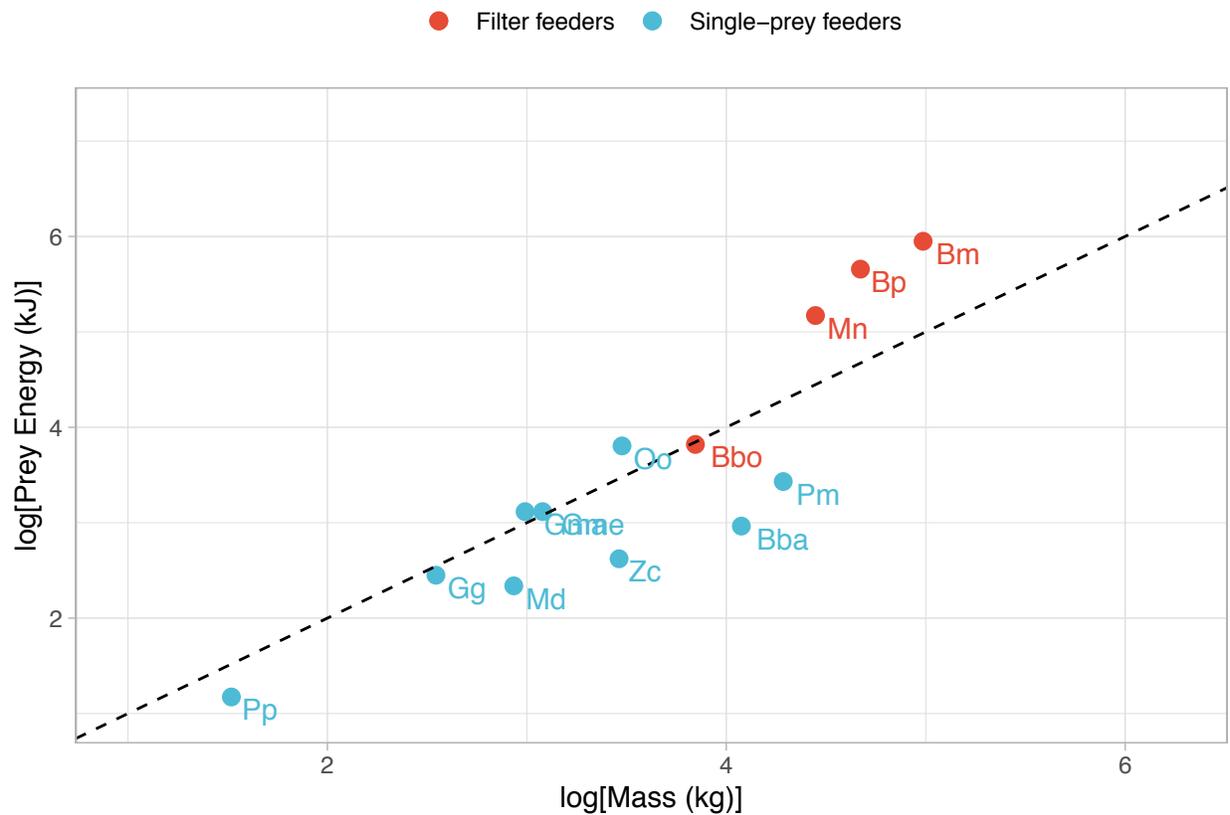
5.5 Plot the data

```
ggplot(smydata, aes(x_mean, y = value, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), shape = 16, size = 3,
            aes(y = y_mean, color = "#4DBBD5FF")) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), shape = 16, size = 3,
            aes(y = y_mean, color = "#E64B35FF")) +
  scale_color_manual(name = "",
                    values = c("#E64B35FF", "#4DBBD5FF"),
                    labels = c("Filter feeders", "Single-prey feeders")) +
  theme_light() + theme(legend.position = "top") +
```

Table 1: Model inputs

species	gr	x_mean	y_mean	fm	abbreviation	Group
Balaenoptera_bonaerensis	1	3.8451	3.8204	Filter	Bbo	Rorqual
Balaenoptera_musculus	1	4.9868	5.9494	Filter	Bm	Rorqual
Balaenoptera_physalus	1	4.6725	5.6574	Filter	Bp	Rorqual
Berardius_bairdii	5	4.0755	2.9647	Single-prey	Bba	Odontocete
Globicephala_macrorhynchus	2	2.9912	3.1162	Single-prey	Gma	Odontocete
Globicephala_melas	2	3.0792	3.1162	Single-prey	Gme	Odontocete
Grampus_griseus	2	2.5441	2.4496	Single-prey	Gg	Odontocete
Megaptera_novaeangliae	1	4.4472	5.1716	Filter	Mn	Rorqual
Mesoplodon_densirostris	5	2.9345	2.3385	Single-prey	Md	Odontocete
Orcinus_orca	2	3.4771	3.8049	Single-prey	Oo	Odontocete
Phocoena_phocoena	3	1.5185	1.1740	Single-prey	Pp	Odontocete
Physeter_macrocephalus	4	4.2856	3.4314	Single-prey	Pm	Odontocete
Ziphius_cavirostris	5	3.4624	2.6223	Single-prey	Zc	Odontocete

```
labs(x = "log[Mass (kg)]", y = "log[Prey Energy (kJ)]") +
geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
ylim(1,7.25) + xlim(1,6.25) +
geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
          aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1) +
geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
          aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1)
```



6 Run OLS with feeding mode as a categorical predictor

6.1 Run OLS and model reduction using ML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "ML")
summary(m.ols)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 20.64824 23.47299 -5.324122
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -3.508870 2.3633173  -1.484722  0.1718
## fmSingle-prey  3.957603 2.4404744   1.621653  0.1393
## x_mean         1.929316 0.5243329   3.679563  0.0051
## fmSingle-prey:x_mean -1.189780 0.5568536  -2.136611  0.0614
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
```

```
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.3664954 -0.7694466 -0.2447162  0.4157708  2.1530605
##
## Residual standard error: 0.364441
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.ols)
```

```
## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)    1 834.3476 <.0001
## fm              1  81.0725 <.0001
## x_mean          1  24.5286 0.0008
## fm:x_mean       1   4.5651 0.0614
```

```
m.ols.2 <- update(m.ols, ~ . - fm:x_mean)
anova(m.ols, m.ols.2)
```

```
##      Model df      AIC      BIC   logLik   Test  L.Ratio p-value
## m.ols      1  5 20.64824 23.47299 -5.324122
## m.ols.2    2  4 23.98184 26.24163 -7.990919 1 vs 2 5.333593 0.0209
```

6.1.1 Compare to an intercept-only model

```
m.ols.0 <- gls(y_mean ~ 1, data = smydata, method = "ML")
anova(m.ols, m.ols.0)
```

```
##      Model df      AIC      BIC   logLik   Test  L.Ratio p-value
## m.ols      1  5 20.64824 23.47299 -5.324122
## m.ols.0    2  2 48.23108 49.36098 -22.115541 1 vs 2 33.58284 <.0001
```

```
m.ols.p <- anova(m.ols, m.ols.0)$`p-value`[2]
```

6.2 Estimate final model using REML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
summary(m.ols)
```

```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC   logLik
## 25.60187 26.58799 -7.800934
##
## Coefficients:
##      Value Std.Error  t-value p-value
## (Intercept) -3.508870 2.3633173 -1.484722 0.1718
## fmSingle-prey 3.957603 2.4404744 1.621653 0.1393
## x_mean        1.929316 0.5243329 3.679563 0.0051
## fmSingle-prey:x_mean -1.189780 0.5568536 -2.136611 0.0614
##
## Correlation:
##      (Intr) fmSng- x_mean
## fmSingle-prey -0.968
```

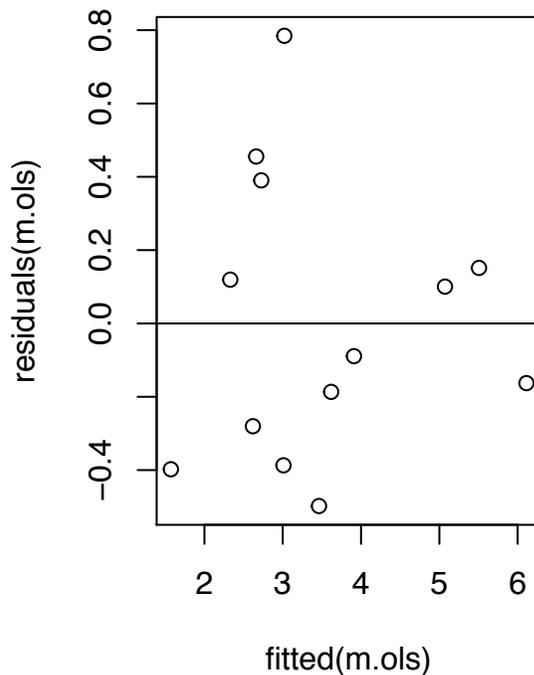
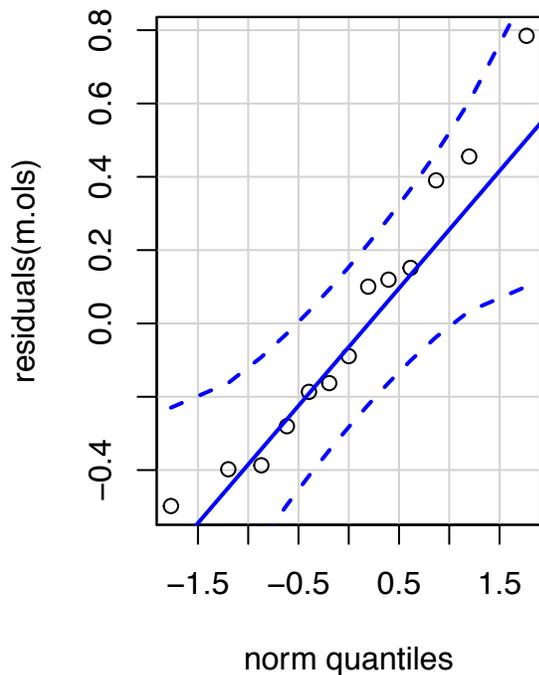
```
## x_mean          -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.1369929 -0.6402182 -0.2036162  0.3459422  1.7914546
##
## Residual standard error: 0.4380036
## Degrees of freedom: 13 total; 9 residual
```

```
m.ols.param <- as.data.frame(t(summary(m.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.ols.param <- m.ols.param[5:8]
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\ppls\\",
             "Figure3_m_ols_param.rds")
saveRDS(m.ols.param,fil)
```

6.2.1 Model diagnostics

6.2.1.1 QQ-plot and Residuals vs fitted plot

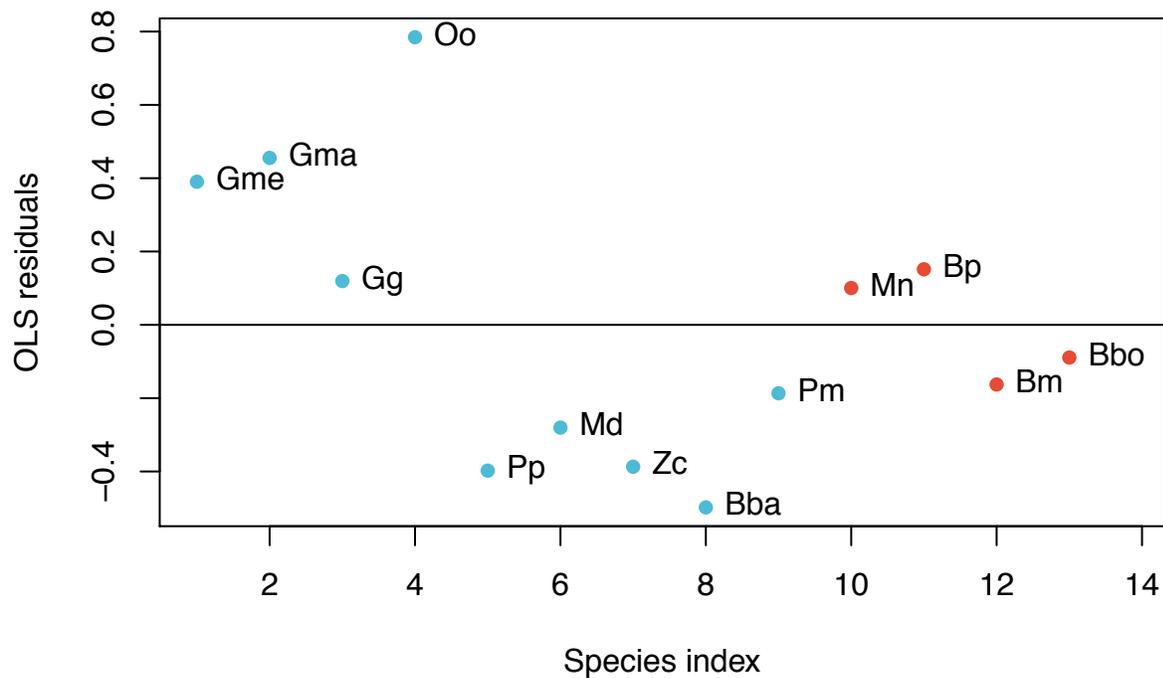
```
par(mfrow=c(1,2))
qqPlot(residuals(m.ols), id=FALSE)
plot(fitted(m.ols), residuals(m.ols))
abline(0,0)
```



6.3 Evaluate for phylogenetic correlation

6.3.1 Plot residuals ordered “by phylogeny” (i.e. in the order of tips of the phylogenetic tree)

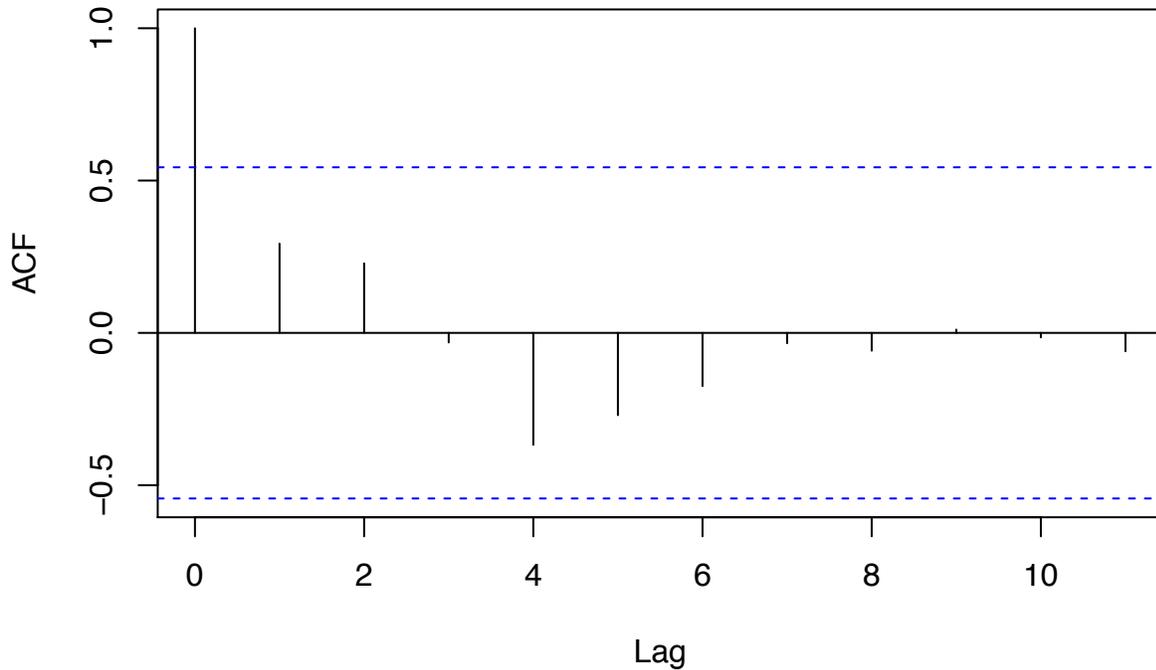
```
is_tip <- smytree$edge[,2]<=length(smytree$tip.label)
ordered_tips <- smytree$edge[is_tip,2] # extract the order of tree tips
oj <- residuals(m.ols)
tl <- smytree$tip.label[ordered_tips]
res <- oj[tl]
plot(oj[tl], pch=16, ylab="OLS residuals", xlab="Species index",
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata[tl, "fm"])],
     xlim=c(1,13.8))
abline(0,0)
text(oj[tl], labels=abbreviation[tl], pos=4)
```



6.3.2 Plot autocorrelation function of residuals ordered “by phylogeny”

```
acf(res, main="Series: residuals sorted by phylogeny")
```

Series: residuals sorted by phylogeny



7 Run pGLS with feeding mode as a categorical predictor

7.1 Estimate Pagel's λ (amount of phylogenetic signal) for each trait separately

Can be informative, but only λ for the entire model should be considered when deciding on whether running pGLS is appropriate.

```
lambdax <- phylosig(smytree, smydata$x_mean, method="lambda", test=T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
lambday <- phylosig(smytree, smydata$y_mean, method="lambda", test=T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
cbind(lambdax, lambday)

##      lambdax      lambday
## lambda 1.014327  1.01842
## logL   -12.69024 -14.32631
## logL0  -17.41681 -22.11554
## P      0.002107892 7.915103e-05
```

7.2 Plot likelihood surface for Pagel's λ for model without feeding mode as a covariate

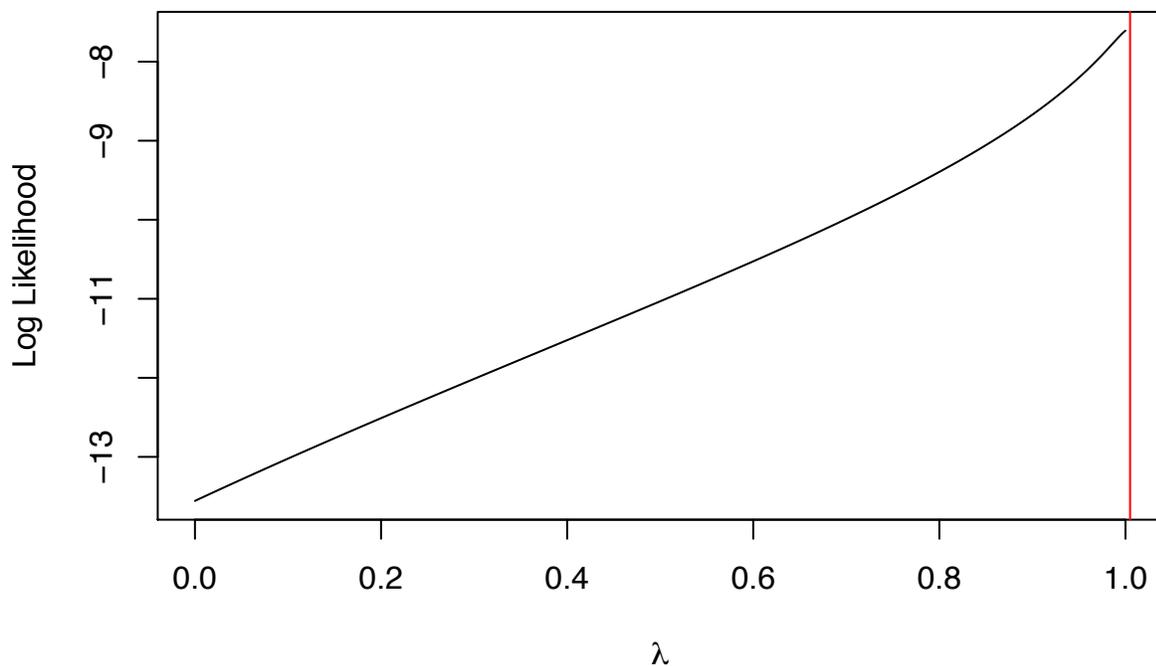
λ estimate for the model marked in red.

```

lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ x_mean, smydata,
  method = "REML", correlation = corPagel(value = lambda, phy = smytree,
  fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
  expression(paste("Prey energy to body mass Likelihood Plot for ", lambda)),
  ylab = "Log Likelihood", xlab = expression(lambda))
m.pa.only <- gls(y_mean ~ x_mean, data = smydata, correlation =
  corPagel(value = 0, phy = smytree, fixed = FALSE), method = "REML")
abline(v = m.pa.only$modelStruct[1], col = "red")

```

Prey energy to body mass Likelihood Plot for λ



7.3 Estimate Pagel's λ for full model using REML

If λ is estimated to be greater than 1, fix it at 1, if smaller than 0, fix it at 0.

$\lambda = 0$ suggests that the relationship between predictor and response variables is unrelated to phylogeny, while $\lambda = 1$ indicates that traits have evolved under Brownian motion on the given phylogeny. Intermediate values of λ indicate that traits have evolved according to a process in which the effect of phylogeny is weaker than in the Brownian model, while values of $\lambda > 1$ can arise if, for instance, traits are more similar than predicted by Brownian motion, given the phylogeny.

```

m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
  corPagel(1, phy = smytree, fixed = FALSE), method = "REML")
summary(m.pgls.nlme)

```

```
## Generalized least squares fit by REML
```

```
## Model: y_mean ~ fm * x_mean
```

```

## Data: smydata
##      AIC      BIC    logLik
## 19.06726 20.2506 -3.533628
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
## 1.00251
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -3.330446 1.5916580 -2.092438 0.0659
## fmSingle-prey  2.813585 1.7028358  1.652294 0.1329
## x_mean        1.883382 0.3482724  5.407784 0.0004
## fmSingle-prey:x_mean -0.904905 0.3841879 -2.355372 0.0429
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey   -0.935
## x_mean          -0.966  0.903
## fmSingle-prey:x_mean 0.875 -0.954 -0.907
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.0667998 -0.2363455 0.2661666 1.0055104 1.9375816
##
## Residual standard error: 0.4745265
## Degrees of freedom: 13 total; 9 residual

```

```
anova(m.pgls.nlme)
```

```

## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)  1 238.13537 <.0001
## fm          1  20.38826 0.0015
## x_mean      1  60.09069 <.0001
## fm:x_mean   1   5.54778 0.0429

```

```

lambda.est <- as.numeric(m.pgls.nlme$modelStruct[1])
if(lambda.est > 1){lambda.est <- 1} else if(lambda.est < 0){lambda.est <- 0}

```

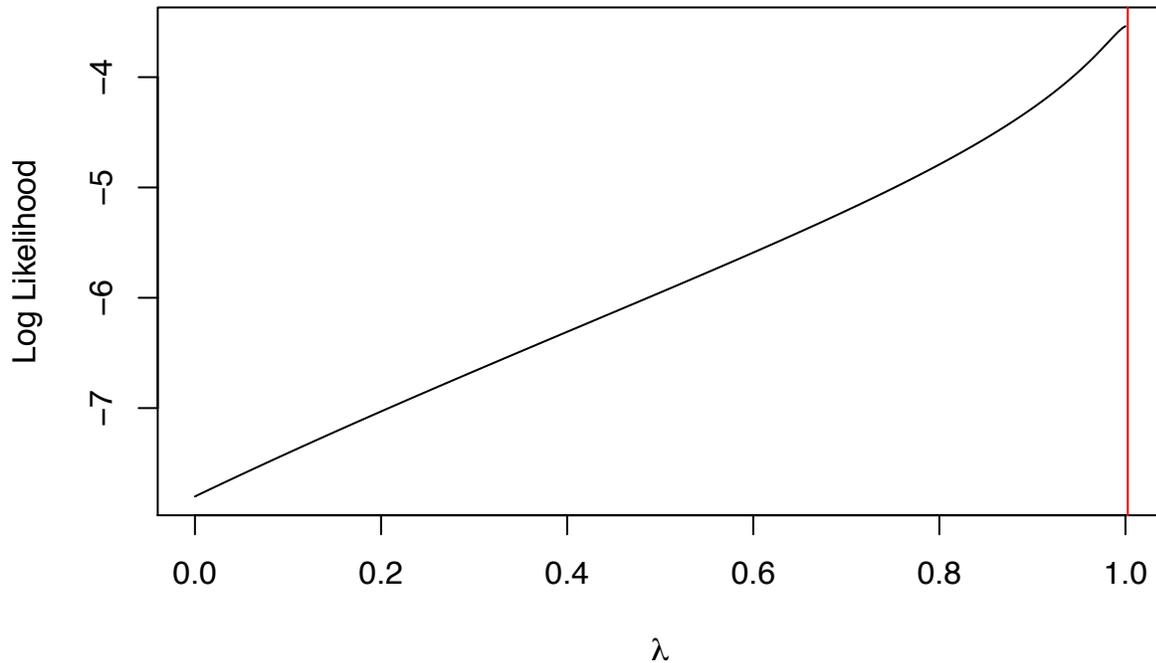
7.4 Plot likelihood surface for Pagel's λ - our estimate marked in red

```

lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ fm * x_mean, smydata,
  method = "REML", correlation = corPagel(value = lambda, phy = smytree,
  fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
  expression(paste("Prey energy to body mass Likelihood Plot for ", lambda)),
  ylab = "Log Likelihood", xlab = expression(lambda))
abline(v = m.pgls.nlme$modelStruct, col = "red")

```

Prey energy to body mass Likelihood Plot for λ



7.5 Run pGLS and model reduction with a fixed Pagel's λ (using ML)

```
m.pgls.nlmf <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
                  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")  
summary(m.pgls.nlmf)
```

```
## Generalized least squares fit by maximum likelihood  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
##      AIC      BIC    logLik  
## 13.38311 16.20786 -1.691557  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
##      1  
##  
## Coefficients:  
##              Value Std.Error   t-value p-value  
## (Intercept)   -3.331445 1.5804760 -2.107875 0.0643  
## fmSingle-prey    2.806201 1.6910301  1.659463 0.1314  
## x_mean          1.883647 0.3458697  5.446117 0.0004  
## fmSingle-prey:x_mean -0.902597 0.3816374 -2.365063 0.0422  
##  
## Correlation:
```

```

##          (Intr) fmSng- x_mean
## fmSingle-prey      -0.935
## x_mean             -0.966  0.903
## fmSingle-prey:x_mean 0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.2993867 -0.2875058 0.3223935 1.2243462 2.3487844
##
## Residual standard error: 0.3912094
## Degrees of freedom: 13 total; 9 residual
anova(m.pgls.nlme)

## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)  1 242.95816 <.0001
## fm           1  20.81370 0.0014
## x_mean       1  61.05413 <.0001
## fm:x_mean    1   5.59352 0.0422

m.pgls.nlme.2 <- update(m.pgls.nlme, ~ . - fm:x_mean)
anova(m.pgls.nlme, m.pgls.nlme.2)

##      Model df      AIC      BIC  logLik  Test  L.Ratio p-value
## m.pgls.nlme      1  5 13.38311 16.20786 -1.691557
## m.pgls.nlme.2    2  4 17.66671 19.92650 -4.833353 1 vs 2 6.283593 0.0122

m.pgls.fm <- gls(y_mean ~ fm, data = smydata, correlation =
corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")
summary(m.pgls.fm)

## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm
## Data: smydata
##      AIC      BIC  logLik
## 37.05832 38.75317 -15.52916
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
## 1
##
## Coefficients:
##      Value Std.Error  t-value p-value
## (Intercept)  4.983250  1.071763  4.649581  0.0007
## fmSingle-prey -2.206898  1.268554 -1.739696  0.1098
##
## Correlation:
##      (Intr)
## fmSingle-prey -0.845
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.4127948 -0.2881020 0.1660987 0.5775873 0.9068152

```

```
##
## Residual standard error: 1.13419
## Degrees of freedom: 13 total; 11 residual
```

```
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 13.38311 16.20786 -1.691557
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -3.331445  1.5804760 -2.107875  0.0643
## fmSingle-prey    2.806201  1.6910301  1.659463  0.1314
## x_mean          1.883647  0.3458697  5.446117  0.0004
## fmSingle-prey:x_mean -0.902597  0.3816374 -2.365063  0.0422
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey   -0.935
## x_mean          -0.966  0.903
## fmSingle-prey:x_mean  0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.2993867 -0.2875058  0.3223935  1.2243462  2.3487844
##
## Residual standard error: 0.3912094
## Degrees of freedom: 13 total; 9 residual
```

7.5.1 Compare to an intercept-only model

```
m.pgls.nlme.0 <- gls(y_mean ~ 1, smydata, correlation = corPagel(value = lambda.est,
  phy = smytree, fixed = TRUE), method = "ML")
anova(m.pgls.nlme, m.pgls.nlme.0)
```

```
##      Model df      AIC      BIC    logLik  Test  L.Ratio
## m.pgls.nlme      1  5 13.38311 16.20786 -1.691557
## m.pgls.nlme.0    2  2 38.21805 39.34794 -17.109023 1 vs 2 30.83493
##      p-value
## m.pgls.nlme
## m.pgls.nlme.0 <.0001
```

```
m.pgls.p <- anova(m.pgls.nlme, m.pgls.nlme.0)$`p-value`[2]
```

7.6 Estimate final model using REML

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "REML")  
summary(m.pgls.nlme)
```

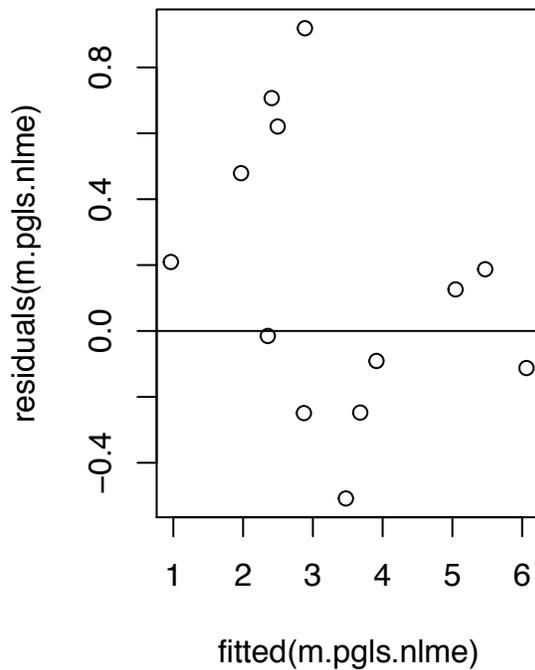
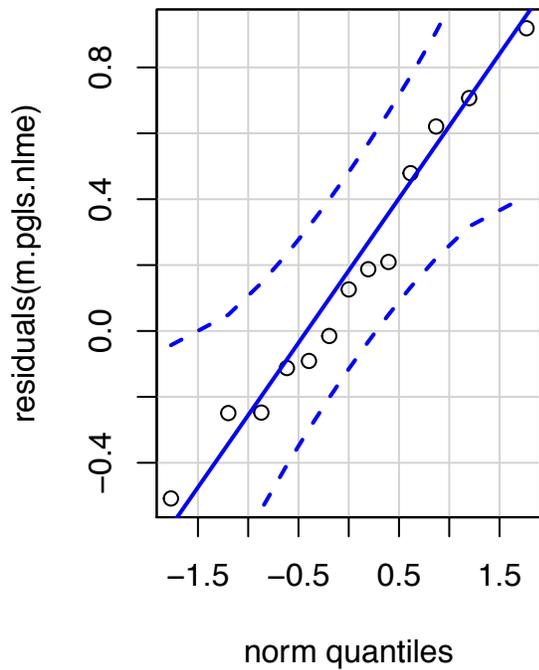
```
## Generalized least squares fit by REML  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
##      AIC      BIC    logLik  
## 17.07736 18.06348 -3.538678  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
##      1  
##  
## Coefficients:  
##              Value Std.Error  t-value p-value  
## (Intercept)   -3.331445  1.5804760 -2.107875  0.0643  
## fmSingle-prey  2.806201  1.6910301  1.659463  0.1314  
## x_mean         1.883647  0.3458697  5.446117  0.0004  
## fmSingle-prey:x_mean -0.902597  0.3816374 -2.365063  0.0422  
##  
## Correlation:  
##              (Intr) fmSng- x_mean  
## fmSingle-prey   -0.935  
## x_mean          -0.966  0.903  
## fmSingle-prey:x_mean  0.875 -0.954 -0.906  
##  
## Standardized residuals:  
##      Min      Q1      Med      Q3      Max  
## -1.0811551 -0.2392192  0.2682476  1.0187176  1.9543067  
##  
## Residual standard error: 0.4701752  
## Degrees of freedom: 13 total; 9 residual
```

```
m.pgls.param <- as.data.frame(t(summary(m.pgls.nlme)$tTable[,1])) %>%  
  mutate(intercept.rorq = `(Intercept)`,  
         intercept.od = `(Intercept)` + `fmSingle-prey`,  
         slope.rorq = `x_mean`, slope.od = `x_mean` + `fmSingle-prey:x_mean`)  
m.pgls.param <- m.pgls.param[5:8]
```

7.6.1 Model diagnostics

7.6.1.1 QQ-plot and Residuals vs fitted plot

```
par(mfrow = c(1,2))  
qqPlot(residuals(m.pgls.nlme), id = FALSE)  
plot(fitted(m.pgls.nlme), residuals(m.pgls.nlme))  
abline(0,0)
```



8 Estimate confidence intervals by bootstrapping

8.1 Bootstrap and compute percentile confidence intervals

```
index <- d_full %>% group_by(Spec) %>% summarize(ix=length(y))
index # number of prey categories for each species
```

```
## # A tibble: 13 x 2
##   Spec          ix
##   <chr>         <int>
## 1 Balaenoptera_bonaerensis     5
## 2 Balaenoptera_musculus       7
## 3 Balaenoptera_physalus       7
## 4 Berardius_bairdii          19
## 5 Globicephala_macrorhynchus  12
## 6 Globicephala_melas        12
## 7 Grampus_griseus            5
## 8 Megaptera_novaeangliae      8
## 9 Mesoplodon_densirostris     3
## 10 Orcinus_orca              12
## 11 Phocoena_phocoena          5
## 12 Physeter_macrocephalus     18
## 13 Ziphius_cavirostris       16
```

```

smydata.orig <- smydata
y_mean <- by(d_full, d_full$Spec, with, weighted.mean(y, Percent))
spec <- spec[match(spec,smydata$species)]

runpGls <- function(smydata,smytree){
  out <- tryCatch(
    {
      model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
        corPagel(lambda.est, phy = smytree, fixed = FALSE),
        method = "REML")
      as.numeric(model.pgls$modelStruct[1])
    },
    error=function(cond) {
      return(NA)
    }
  )
}

a.ols <- matrix(nrow=10000,ncol=4)
a.pgls <- matrix(nrow=10000,ncol=4)
b <- matrix(nrow=10000,ncol=length(spec))
boot.lambdas <- rep(NA,10000)
for(i in 1:10000){
  for(sp in 1:length(spec)){
    ix <- sample(1:index$ix[index$Spec==spec[sp]], replace = T)
    y_mean[sp] <- sum(d_full[d_full$Spec==spec[sp],"y"][ix]*
      d_full[d_full$Spec==spec[sp],"Percent"][ix])/
      sum(d_full[d_full$Spec==spec[sp],"Percent"][ix])
  }
  smydata$y_mean <- y_mean

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
  myout <- runpGls(smydata,smytree)
  boot.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(lambda.est, phy = smytree, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(l.est, phy = smytree, fixed = TRUE), method = "REML")
  }

  a.ols[i,] <- c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4])
  a.pgls[i,] <- c(coef(model.pgls)[1],coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],coef(model.pgls)[3]+coef(model.pgls)[4])
  b[i,] <- predict(model.ols)
}

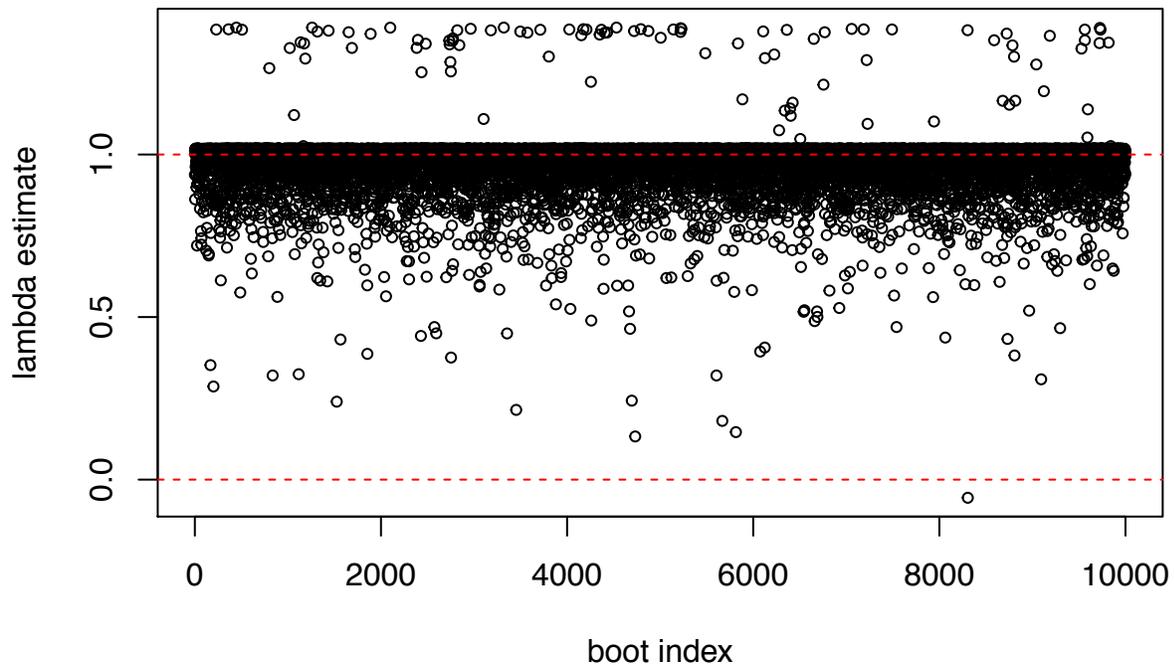
# number of pGLS models, where lambda could not be estimated ==> used original value:

```

```
sum(is.na(boot.lambdas))
```

```
## [1] 271
```

```
plot(boot.lambdas, cex=.7, xlab="boot index", ylab="lambda estimate")  
abline(h=0,lty="dashed",col="red")  
abline(h=1,lty="dashed",col="red")
```



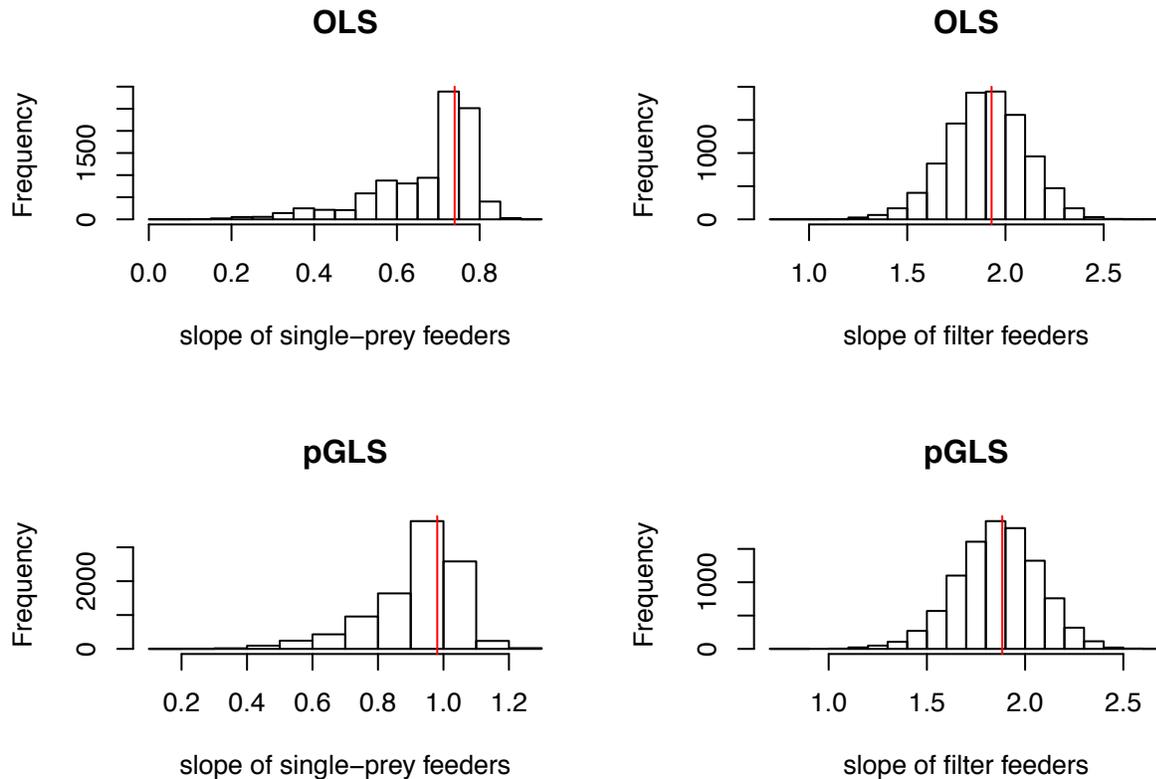
```
preds <- apply(b, 2, quantile, c(0.025, 0.975))  
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",  
             "Figure3_bootstrap_b.rds")  
saveRDS(b,fil)  
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",  
             "Figure3_bootstrap_preds.rds")  
saveRDS(preds,fil)  
  
df.boot.ols <- data.frame(cbind(t(m.ols.param),t(t(apply(a.ols, 2, mean))),  
                             t(apply(a.ols, 2, quantile, c(0.025, 0.975)))))  
names(df.boot.ols) <- c("obs","bootest","lowerCI","upperCI")  
df.boot.pgls <- data.frame(cbind(t(m.pgls.param),t(t(apply(a.pgls, 2, mean))),  
                              t(apply(a.pgls, 2, quantile, c(0.025, 0.975)))))  
names(df.boot.pgls) <- c("obs","bootest","lowerCI","upperCI")  
  
par(mfrow=c(2,2))  
hist(a.ols[,4], xlab="slope of single-prey feeders", main="OLS")  
abline(v=m.ols.param[4], col="red")  
hist(a.ols[,3], xlab="slope of filter feeders", main="OLS")
```

```

abline(v=m.ols.param[3], col="red")

hist(a.pgls[,4], xlab="slope of single-prey feeders", main="pGLS")
abline(v=m.pgls.param[4], col="red")
hist(a.pgls[,3], xlab="slope of filter feeders", main="pGLS")
abline(v=m.pgls.param[3], col="red")

```



8.2 Compute BCa (bias-corrected and accelerated) confidence intervals

Corrects for bias and skewness in the distribution of bootstrap estimates.

Based on <https://blogs.sas.com/content/iml/2017/07/12/bootstrap-bca-interval.html>

```

smydata <- smydata.orig

# compute bias-correction factor from the proportion of bootstrap estimates
# that are less than the observed estimate

bootBC <- function(bootEst, Est){
  B <- ncol(bootEst)*nrow(bootEst) # number of bootstrap samples
  propLess <- sum(bootEst < Est)/B # proportion of replicates less than observed stat
  z0 <- qnorm(propLess) # bias correction
  return(z0)
}

z0.ols <- numeric()
for (i in 1:ncol(a.ols)){

```

```

z0.ols[i] <- bootBC(t(t(a.ols[,i])),as.numeric(m.ols.param[i]))
}

z0.pgls <- numeric()
for (i in 1:ncol(a.pgls)){
z0.pgls[i] <- bootBC(t(t(a.pgls[,i])),as.numeric(m.pgls.param[i]))
}

# compute acceleration factor, which is related to the skewness of bootstrap estimates.
# Use jackknife replicates to estimate.

jStat.ols <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jStat.pgls <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jack.lambdas <- rep(NA,nrow(smydata))
for (i in 1:nrow(smydata)) {
  d_sub <- subset(d_full, Spec==smydata$species[i])
  y_mean.j <- numeric()
  for(j in 1:nrow(d_sub)){
    d_sub.j <- d_sub[-j,]
    y_mean.j[j] <- sum(d_sub.j$y*d_sub.j$Percent)/sum(d_sub.j$Percent)
  }
  smydata.j <- smydata
  smydata.j$y_mean[i] <- mean(y_mean.j)
  pruned.tree <- drop.tip(smytree,smytree$tip.label[-match(smydata.j$species,
    smytree$tip.label)])

  smytree.j <- pruned.tree
  smydata.j <- smydata.j[match(smytree.j$tip.label,rownames(smydata.j)),]

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata.j, method = "REML")

  myout <- runpGls(smydata.j,smytree.j)
  jack.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(lambda.est, phy = smytree.j, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(l.est, phy = smytree.j, fixed = TRUE), method = "REML")
  }

  jStat.ols[,i] <- as.numeric(c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4]))
  jStat.pgls[,i] <- as.numeric(c(coef(model.pgls)[1],
    coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],
    coef(model.pgls)[3]+coef(model.pgls)[4]))
}
jackEst.ols <- t(t(apply(jStat.ols, 1, mean))) # jackknife estimate
jackEst.pgls <- t(t(apply(jStat.pgls, 1, mean))) # jackknife estimate

```

```

jack.lambdas # lambdas of the jackknifed models

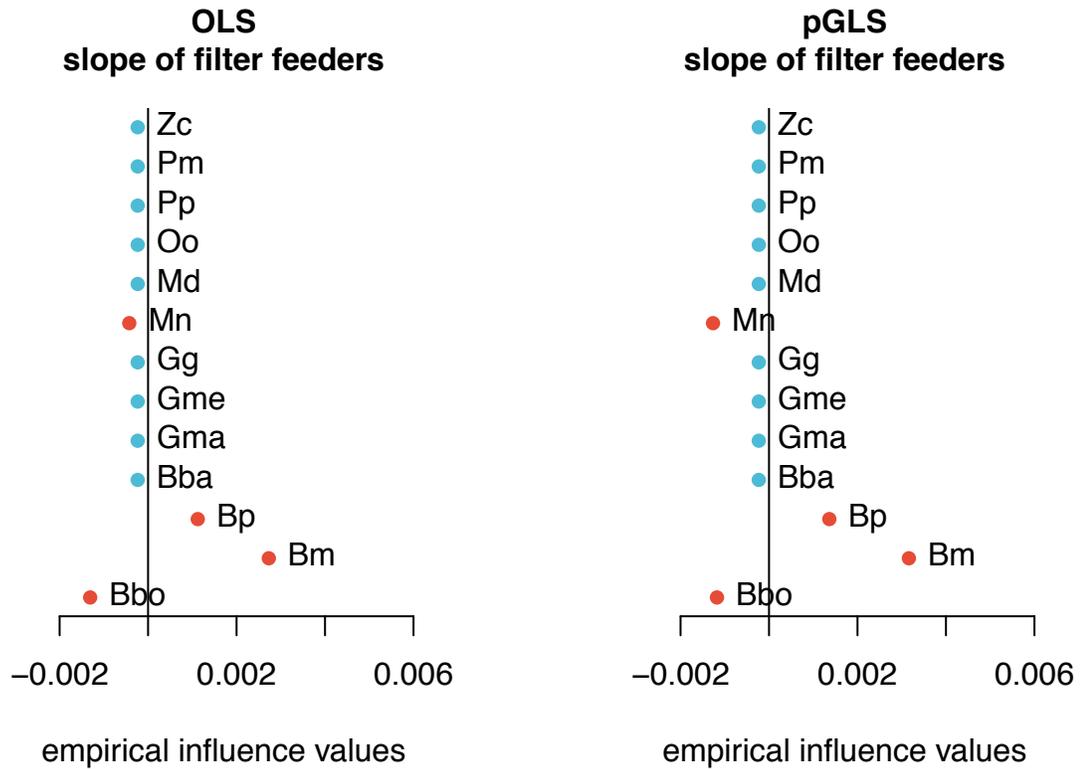
## [1] 1.002515 1.002540 1.002476 1.002547 1.002256 1.002764 1.002529
## [8] 1.002497 1.002599 1.002496 1.002818 1.002511 1.002507

num.ols <- numeric(); den.ols <- numeric(); ahat.ols <- numeric()
num.pgls <- numeric(); den.pgls <- numeric(); ahat.pgls <- numeric()
for (i in 1:nrow(jStat.ols)){
  num.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^3 )
  den.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^2 )
  ahat.ols[i] <- num.ols[i]/(6*den.ols[i]^(3/2)) # ahat based on jackknife
  num.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^3 )
  den.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^2 )
  ahat.pgls[i] <- num.pgls[i]/(6*den.pgls[i]^(3/2)) # ahat based on jackknife
}

# influential species:
par(mfrow=c(1,2))
plot(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("OLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("pGLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```

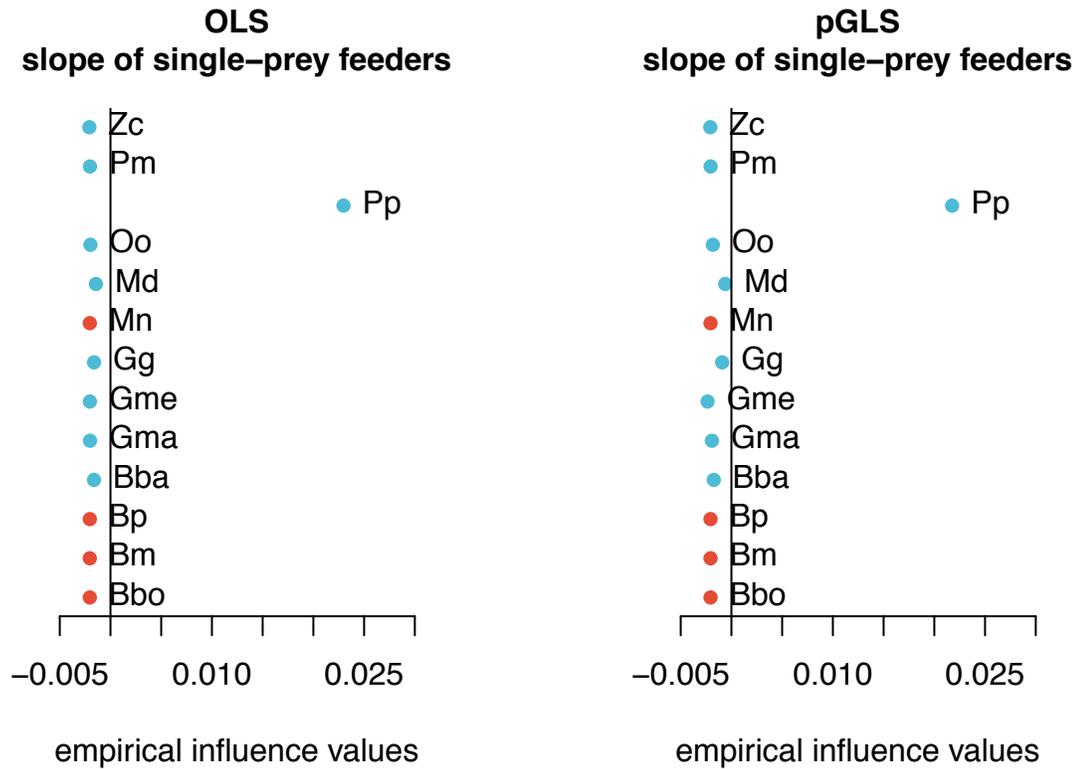


```

par(mfrow=c(1,2))
plot(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("OLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("pGLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```



```
# adjust quantiles for 100*(1-alpha)% bootstrap BCa interval
```

```
alpha <- 0.05
zL.ols <- z0.ols + qnorm(alpha/2)
alpha1.ols <- pnorm(z0.ols + zL.ols / (1-ahat.ols*zL.ols))
zU.ols <- z0.ols + qnorm(1-alpha/2)
alpha2.ols <- pnorm(z0.ols + zU.ols / (1-ahat.ols*zU.ols))

zL.pgls <- z0.pgls + qnorm(alpha/2)
alpha1.pgls <- pnorm(z0.pgls + zL.pgls / (1-ahat.pgls*zL.pgls))
zU.pgls <- z0.pgls + qnorm(1-alpha/2)
alpha2.pgls <- pnorm(z0.pgls + zU.pgls / (1-ahat.pgls*zU.pgls))

cbind((alpha1.ols*100), (alpha2.ols*100)) # new quantiles OLS
```

```
##           [,1]      [,2]
## [1,]  0.498972967  93.54027
## [2,]  0.009319683  84.25180
## [3,]  6.804385905  99.56620
## [4,] 16.419165418  99.99256
```

```
cbind((alpha1.pgls*100), (alpha2.pgls*100)) # new quantiles pGLS
```

```
##           [,1]      [,2]
## [1,]  0.66575532  94.29030
## [2,]  0.01137115  84.73977
## [3,]  6.02055602  99.42859
```

```
## [4,] 15.98149605 99.99108
CI.ols <- matrix(nrow = ncol(a.ols), ncol=2)
for (i in 1:ncol(a.ols)){
CI.ols[i,] <- quantile(a.ols[,i], c(alpha1.ols[i], alpha2.ols[i])) # BCa interval
}
df.boot.ols$lowerCIbca <- CI.ols[,1]
df.boot.ols$upperCIbca <- CI.ols[,2]

CI.pgls <- matrix(nrow = ncol(a.pgls), ncol=2)
for (i in 1:ncol(a.pgls)){
CI.pgls[i,] <- quantile(a.pgls[,i], c(alpha1.pgls[i], alpha2.pgls[i])) # BCa interval
}
df.boot.pgls$lowerCIbca <- CI.pgls[,1]
df.boot.pgls$upperCIbca <- CI.pgls[,2]
```

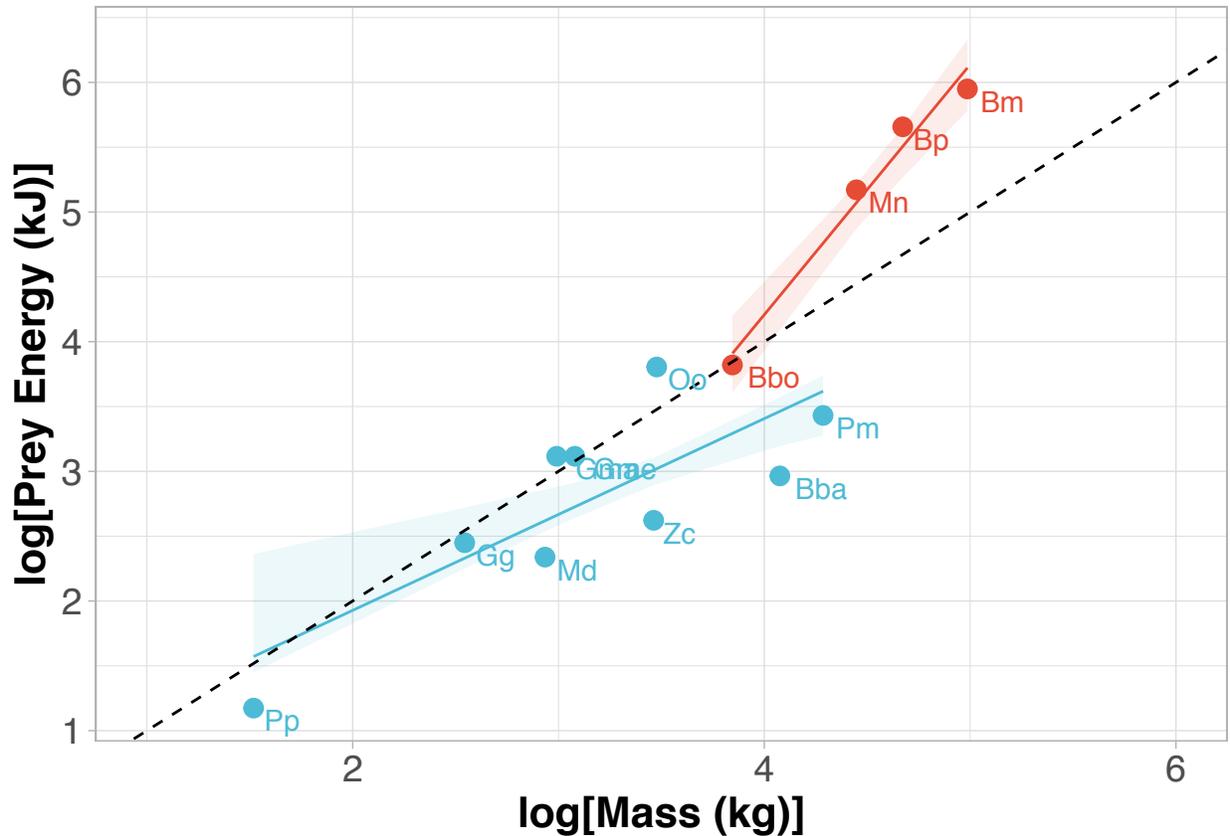
8.3 Plot OLS model

```
smydata <- smydata.orig

ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit,
                                     y_min = preds[1,], y_max = preds[2,]))

fig_ols <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
            linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
            linetype = 1) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Rorqual"), fill = "#E64B35FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Odontocete"), fill = "#4DBBD5FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Prey Energy (kJ)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_ols
```



8.3.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCI"],
  lowerCI.od = df.boot.ols["slope.od","lowerCI"],
  upperCI.od = df.boot.ols["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.ols["slope.od","lowerCIbca"],
  upperCI.od = df.boot.ols["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.ols[,3],
  odontocete_slope=a.ols[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF", color = NA, alpha = 0.3) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF", color = NA, alpha = 0.3) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 1, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),

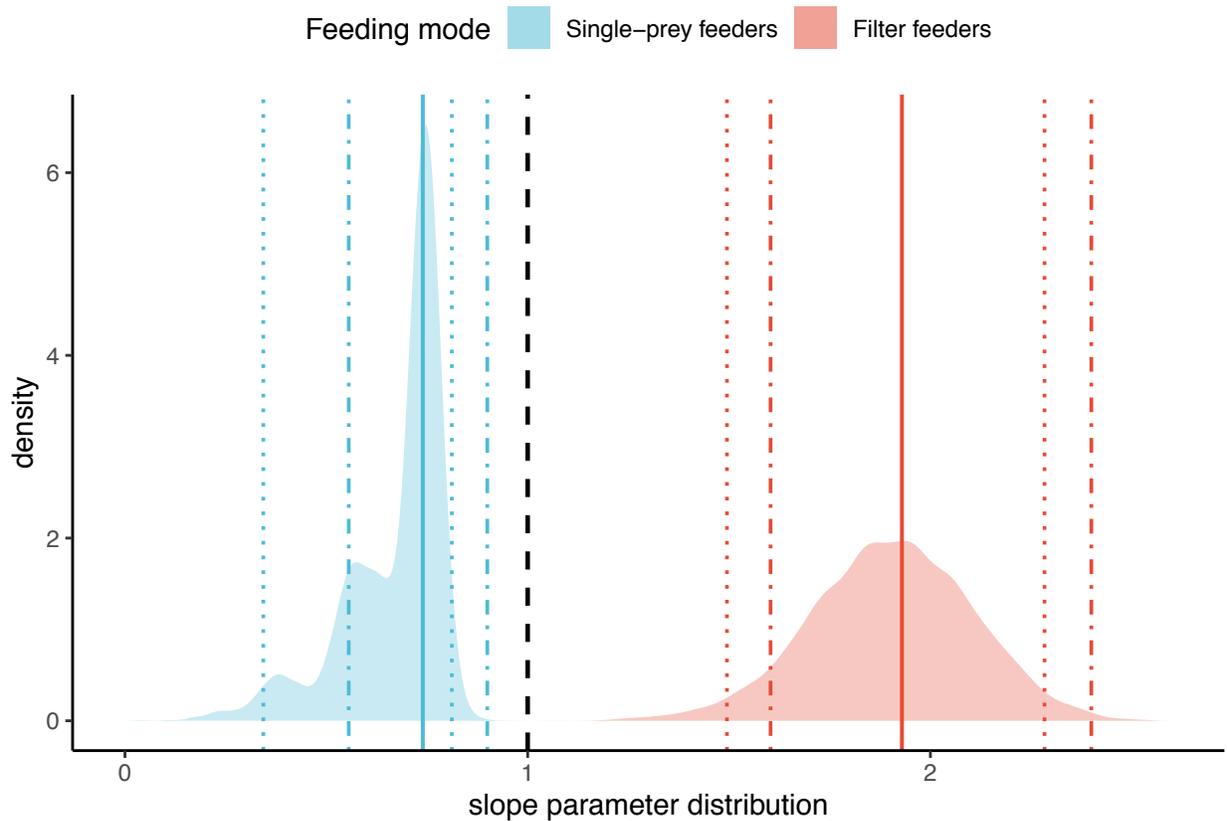
```

```

    color = "#4DBBD5FF", linetype=1, size = 0.7) +
  geom_vline(data=model_param, aes(xintercept=lowerCI.od),
    color = "#4DBBD5FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param, aes(xintercept=upperCI.od),
    color = "#4DBBD5FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param, aes(xintercept=slope.rorq),
    color = "#E64B35FF", linetype=1, size = 0.7) +
  geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
    color = "#E64B35FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
    color = "#E64B35FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
    color = "#E64B35FF", linetype=4, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
    color = "#E64B35FF", linetype=4, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
    color = "#4DBBD5FF", linetype=4, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
    color = "#4DBBD5FF", linetype=4, size = 0.65) +
  xlim(0,2.6) +
  theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 4 rows containing non-finite values (stat_density).



```
rn <- rownames(df.boot.ols)
rownames(df.boot.ols) <- c("intercept filter","intercept single-prey",
                          "slope filter","slope single-prey")
knitr::kable(df.boot.ols,
             caption = "OLS 95%\%$ Bootstrap Pctl and BCa CI",
             format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
  kable_styling(latex_options = "hold_position")
```

Table 2: OLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-3.5089	-3.4119	-5.1333	-1.6020	-5.6060	-2.0407
intercept single-prey	0.4487	0.6826	0.2296	1.8407	-0.0428	1.1061
slope filter	1.9293	1.9037	1.4948	2.2833	1.6031	2.3996
slope single-prey	0.7395	0.6715	0.3436	0.8116	0.5556	0.8998

```
rownames(df.boot.ols) <- rn
```

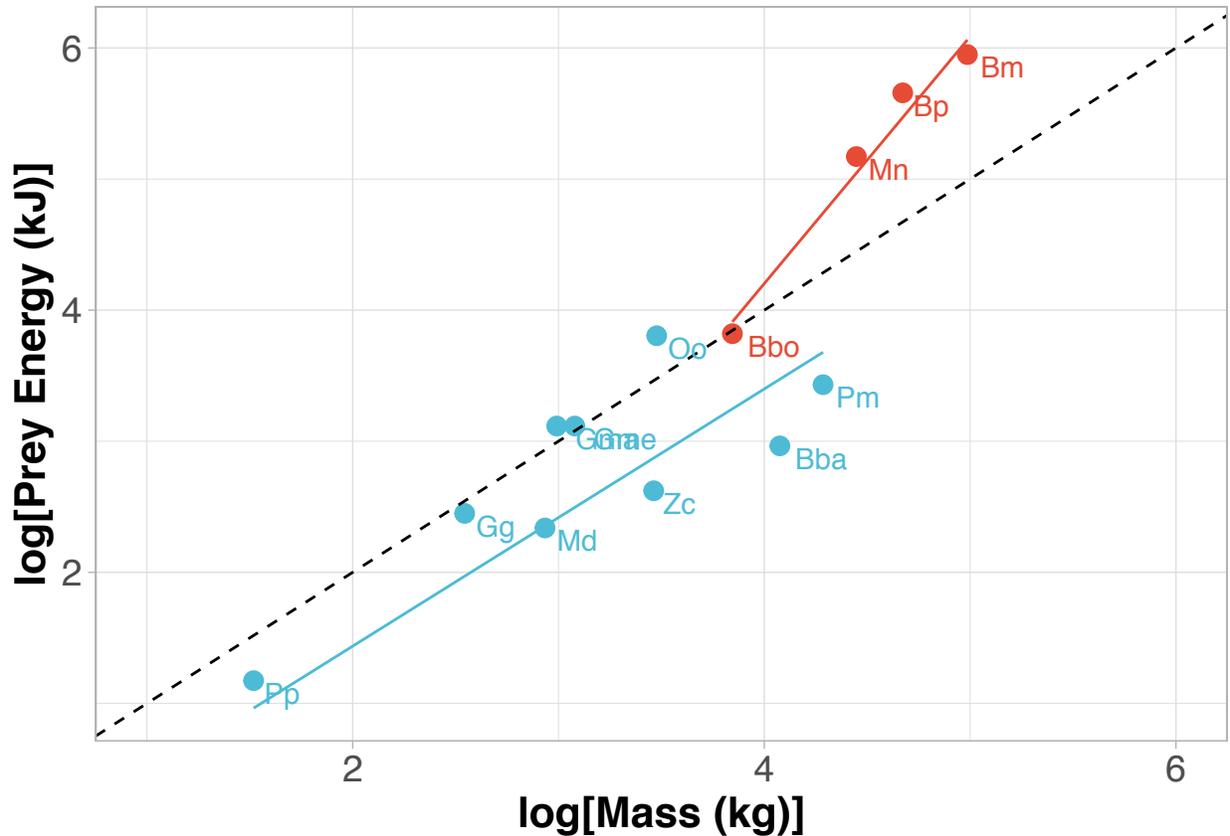
8.4 Plot pGLS model

```
smydata <- smydata.orig

pgls.fit <- predict(m.pgls.nlme)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))

fig_pgls <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
            linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
            linetype = 1) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Prey Energy (kJ)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_pgls
```



8.4.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.pglis["slope.rorq","obs"],
  slope.od = df.boot.pglis["slope.od","obs"],
  lowerCI.rorq = df.boot.pglis["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.pglis["slope.rorq","upperCI"],
  lowerCI.od = df.boot.pglis["slope.od","lowerCI"],
  upperCI.od = df.boot.pglis["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.pglis["slope.rorq","obs"],
  slope.od = df.boot.pglis["slope.od","obs"],
  lowerCI.rorq = df.boot.pglis["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.pglis["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.pglis["slope.od","lowerCIbca"],
  upperCI.od = df.boot.pglis["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.pglis[,3],
  odontocete_slope=a.pglis[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF"), color = NA, alpha = 0.3) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF"), color = NA, alpha = 0.3) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 1, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),

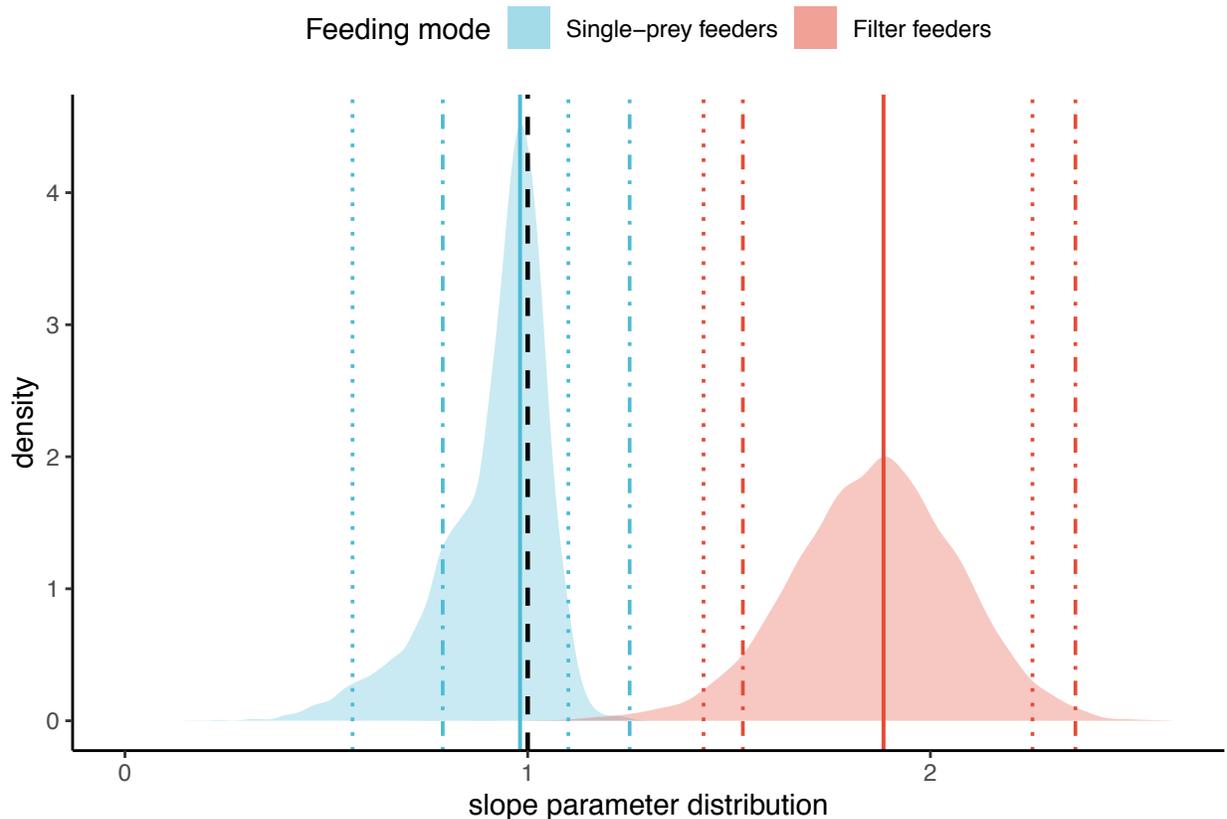
```

```

    color = "#4DBBD5FF", linetype=1, size = 0.7) +
  geom_vline(data=model_param, aes(xintercept=lowerCI.od),
    color = "#4DBBD5FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param, aes(xintercept=upperCI.od),
    color = "#4DBBD5FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param, aes(xintercept=slope.rorq),
    color = "#E64B35FF", linetype=1, size = 0.7) +
  geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
    color = "#E64B35FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
    color = "#E64B35FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
    color = "#E64B35FF", linetype=4, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
    color = "#E64B35FF", linetype=4, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
    color = "#4DBBD5FF", linetype=4, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
    color = "#4DBBD5FF", linetype=4, size = 0.65) +
  xlim(0,2.6) +
  theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 2 rows containing non-finite values (stat_density).



```
rn <- rownames(df.boot.pgls)
rownames(df.boot.pgls) <- c("intercept filter","intercept single-prey",
    "slope filter","slope single-prey")
knitr::kable(df.boot.pgls,
    caption = "pGLS 95%\% Bootstrap Pctl and BCa CI",
    format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
    kable_styling(latex_options = "hold_position")
```

Table 3: pGLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-3.3314	-3.2483	-5.0311	-1.3635	-5.4463	-1.7731
intercept single-prey	-0.5252	-0.2837	-0.9187	1.0506	-1.4802	0.2083
slope filter	1.8836	1.8618	1.4367	2.2534	1.5343	2.3601
slope single-prey	0.9811	0.9171	0.5651	1.1010	0.7891	1.2532

```
rownames(df.boot.pgls) <- rn
```

9 Extract summary statistics

```
specify_decimal <- function(x, k) trimws(format(round(x, k), nsmall = k))

res.df.ols <- m.ols$dim$N - m.ols$dim$p

res.df.pgls <- m.pgls.nlme$dim$N - m.pgls.nlme$dim$p

intercepts.od.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.od","obs"],4),
    " (", specify_decimal(df.boot.pgls["intercept.od","lowerCI"],4),
    " - ", specify_decimal(df.boot.pgls["intercept.od","upperCI"],4),
    ")"),
    paste0(specify_decimal(df.boot.pgls["intercept.od","obs"],4),
    " (", specify_decimal(df.boot.pgls["intercept.od","lowerCIbca"],4),
    " - ", specify_decimal(df.boot.pgls["intercept.od","upperCIbca"],4),
    ")"),
    paste0(specify_decimal(df.boot.ols["intercept.od","obs"],4),
    " (", specify_decimal(df.boot.ols["intercept.od","lowerCI"],4),
    " - ", specify_decimal(df.boot.ols["intercept.od","upperCI"],4),
    ")"),
    paste0(specify_decimal(df.boot.ols["intercept.od","obs"],4),
    " (", specify_decimal(df.boot.ols["intercept.od","lowerCIbca"],4),
    " - ", specify_decimal(df.boot.ols["intercept.od","upperCIbca"],4),
    ")"))

intercepts.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.rorq","obs"],4),
    " (", specify_decimal(df.boot.pgls["intercept.rorq","lowerCI"],4),
    " - ", specify_decimal(df.boot.pgls["intercept.rorq","upperCI"],4),
    ")"),
    paste0(specify_decimal(df.boot.pgls["intercept.rorq","obs"],4),
    " (", specify_decimal(df.boot.pgls["intercept.rorq","lowerCIbca"],4),
    " - ", specify_decimal(df.boot.pgls["intercept.rorq","upperCIbca"],
    4),")"),
    paste0(specify_decimal(df.boot.ols["intercept.rorq","obs"],4),
```

```

      " (", specify_decimal(df.boot.ols["intercept.rorq","lowerCI"],4),
      " - ", specify_decimal(df.boot.ols["intercept.rorq","upperCI"],4),
      ")"),
    paste0(specify_decimal(df.boot.ols["intercept.rorq","obs"],4),
      " (", specify_decimal(df.boot.ols["intercept.rorq","lowerCIbca"],4),
      " - ", specify_decimal(df.boot.ols["intercept.rorq","upperCIbca"],4),
      ")"))

slopes.od.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.od","obs"],4)," (",
  specify_decimal(df.boot.pgl["slope.od","lowerCI"],4)," - ",
  specify_decimal(df.boot.pgl["slope.od","upperCI"],4),")"),
  paste0(specify_decimal(df.boot.pgl["slope.od","obs"],4)," (",
  specify_decimal(df.boot.pgl["slope.od","lowerCIbca"],4)," - ",
  specify_decimal(df.boot.pgl["slope.od","upperCIbca"],4),")"),
  paste0(specify_decimal(df.boot.ols["slope.od","obs"],4)," (",
  specify_decimal(df.boot.ols["slope.od","lowerCI"],4)," - ",
  specify_decimal(df.boot.ols["slope.od","upperCI"],4),")"),
  paste0(specify_decimal(df.boot.ols["slope.od","obs"],4)," (",
  specify_decimal(df.boot.ols["slope.od","lowerCIbca"],4)," - ",
  specify_decimal(df.boot.ols["slope.od","upperCIbca"],4),")"))

slopes.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.rorq","obs"],4)," (",
  specify_decimal(df.boot.pgl["slope.rorq","lowerCI"],4)," - ",
  specify_decimal(df.boot.pgl["slope.rorq","upperCI"],4),")"),
  paste0(specify_decimal(df.boot.pgl["slope.rorq","obs"],4)," (",
  specify_decimal(df.boot.pgl["slope.rorq","lowerCIbca"],4),
  " - ", specify_decimal(df.boot.pgl["slope.rorq","upperCIbca"],
  4),")"),
  paste0(specify_decimal(df.boot.ols["slope.rorq","obs"],4)," (",
  specify_decimal(df.boot.ols["slope.rorq","lowerCI"],4)," - ",
  specify_decimal(df.boot.ols["slope.rorq","upperCI"],4),")"),
  paste0(specify_decimal(df.boot.ols["slope.rorq","obs"],4)," (",
  specify_decimal(df.boot.ols["slope.rorq","lowerCIbca"],4),
  " - ", specify_decimal(df.boot.ols["slope.rorq","upperCIbca"],
  4),")"))

a.od.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.od","obs"]),4)," (",
  specify_decimal(10^(df.boot.pgl["intercept.od","lowerCI"]),4),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.od","upperCI"]),4),
  ")"),
  paste0(specify_decimal(10^(df.boot.pgl["intercept.od","obs"]),4)," (",
  specify_decimal(10^(df.boot.pgl["intercept.od","lowerCIbca"]),4),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.od","upperCIbca"]),4),
  ")"),
  paste0(specify_decimal(10^(df.boot.ols["intercept.od","obs"]),4)," (",
  specify_decimal(10^(df.boot.ols["intercept.od","lowerCI"]),4)," - ",
  specify_decimal(10^(df.boot.ols["intercept.od","upperCI"]),4),")"),
  paste0(specify_decimal(10^(df.boot.ols["intercept.od","obs"]),4)," (",
  specify_decimal(10^(df.boot.ols["intercept.od","lowerCIbca"]),4),
  " - ", specify_decimal(10^(df.boot.ols["intercept.od","upperCIbca"]),
  4),")"))

a.rorq.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq","obs"]),4),
  " (", specify_decimal(10^(df.boot.pgl["intercept.rorq","lowerCI"]),5),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.rorq","upperCI"]),4),")"),

```

Table 4: Model summary statistics

	<i>Filter feeders</i>		<i>Single-prey feeders</i>		RSE	tot.df	res.df
	slope*	intercept	slope	intercept			
pGLS	1.8836 (1.4367 - 2.2534)	-3.3314 (-5.0311 - -1.3635)	0.9811 (0.5651 - 1.1010)	-0.5252 (-0.9187 - 1.0506)	0.4702	13	9
	1.8836 (1.5343 - 2.3601)	-3.3314 (-5.4463 - -1.7731)	0.9811 (0.7891 - 1.2532)	-0.5252 (-1.4802 - 0.2083)			
OLS	1.9293 (1.4948 - 2.2833)	-3.5089 (-5.1333 - -1.6020)	0.7395 (0.3436 - 0.8116)	0.4487 (0.2296 - 1.8407)	0.4380		
	1.9293 (1.6031 - 2.3996)	-3.5089 (-5.6060 - -2.0407)	0.7395 (0.5556 - 0.8998)	0.4487 (-0.0428 - 1.1061)			

Note:

* Throughout the table, values in brackets represent 95% confidence intervals: percentile in shaded rows, BCa in non-shaded rows.

```

paste0(specify_decimal(10^(df.boot.pgls["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.pgls["intercept.rorq","lowerCIbca"]),5),
" - ", specify_decimal(10^(df.boot.pgls["intercept.rorq","upperCIbca"]),4),
")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCI"]),5),
" - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCI"]),4),")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCIbca"]),5),
" - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCIbca"]),4),
")"))

RSE <- rbind(specify_decimal(t(t(rep(as.numeric(m.pgls.nlm$sigma),2))),4),
specify_decimal(t(t(rep(as.numeric(m.ols$sigma),2))),4))
df <- cbind(t(t(c(rep(m.pgls.nlm$dims$N,2), rep(m.ols$dims$N,2)))),
t(t(c(rep(res.df.pgls,2), rep(res.df.ols,2)))))
models <- rbind(t(t(rep("pGLS",2))),t(t(rep("OLS",2))))

outputs <- cbind(models, slopes.rorq.ci, intercepts.rorq.ci, slopes.od.ci,
intercepts.od.ci, RSE, df)
df.outputs <- data.frame(outputs, check.rows = TRUE, check.names = TRUE)
names(df.outputs) <- c("","slope","intercept","slope","intercept","RSE", "tot.df","res.df")
names(df.outputs)[2] <- paste0(names(df.outputs)[2],
footnote_marker_symbol(1))

knitr::kable(df.outputs,
caption = "Model summary statistics",
format = "latex", booktabs = TRUE, escape = FALSE) %>%
kable_styling(latex_options = "scale_down") %>%
row_spec(0, bold = T) %>%
row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
column_spec(c(1,(ncol(df.outputs)-1):ncol(df.outputs))-1,
background = "white") %>%
column_spec(1, bold = T) %>%
collapse_rows(columns = c(1,(ncol(df.outputs)-2):ncol(df.outputs))) %>%
add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2,
" " = 3), bold = T, italic = T) %>%
footnote(general = "", general_title = "Note:",
symbol = paste0("Throughout the table, values in brackets",
" represent 95% confidence intervals: ",
"percentile in shaded rows, BCa in non-shaded rows."),
symbol_title = "", title_format = "italic",
footnote_as_chunk = T)

```

Table 5: Transformed to allometric equations

	<i>Filter feeders</i>		<i>Single-prey feeders</i>	
	a*	b	a	b
pGLS	5e-04 (1e-05 - 0.0433)	1.8836 (1.4367 - 2.2534)	0.2984 (0.1206 - 11.2365)	0.9811 (0.5651 - 1.1010)
	5e-04 (0.00000 - 0.0169)	1.8836 (1.5343 - 2.3601)	0.2984 (0.0331 - 1.6153)	0.9811 (0.7891 - 1.2532)
OLS	3e-04 (1e-05 - 0.0250)	1.9293 (1.4948 - 2.2833)	2.8102 (1.6966 - 69.2974)	0.7395 (0.3436 - 0.8116)
	3e-04 (0.00000 - 0.0091)	1.9293 (1.6031 - 2.3996)	2.8102 (0.9062 - 12.7687)	0.7395 (0.5556 - 0.8998)

* Throughout the table, values in brackets represent 95% confidence intervals.: percentile in shaded rows, BCa in non-shaded rows.

```
alloout <- cbind(models, a.rorq.ci, slopes.rorq.ci, a.od.ci, slopes.od.ci)
df.allo <- data.frame(alloout, check.rows = TRUE, check.names = TRUE)
names(df.allo) <- c("", "a", "b", "a", "b")
names(df.allo)[2] <- paste0(names(df.allo)[2], footnote_marker_symbol(1))
knitr::kable(df.allo,
  caption = "Transformed to allometric equations",
  format = "latex", booktabs = TRUE, escape = FALSE) %>%
  kable_styling(latex_options = "scale_down") %>%
  row_spec(0, bold = T) %>%
  row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
  column_spec(1, bold = T) %>%
  collapse_rows(columns = 1) %>%
  add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2),
    bold = T, italic = T) %>%
  footnote(symbol = paste0("Throughout the table, values in brackets",
    " represent 95% confidence intervals.: ",
    "percentile in shaded rows, BCa in non-shaded rows."),
    symbol_title = "", threeparttable = TRUE, footnote_as_chunk = T)
```

10 Plot best models (OLS - dashed, PGLS - solid)

```
pgls.fit <- predict(m.pgls.nlme)
ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))

fig_3 <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe2, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 2) +
  geom_line(data = dplyr::filter(predframe2, Group == "Odontocete"), color = "#4DBBD5FF",
```

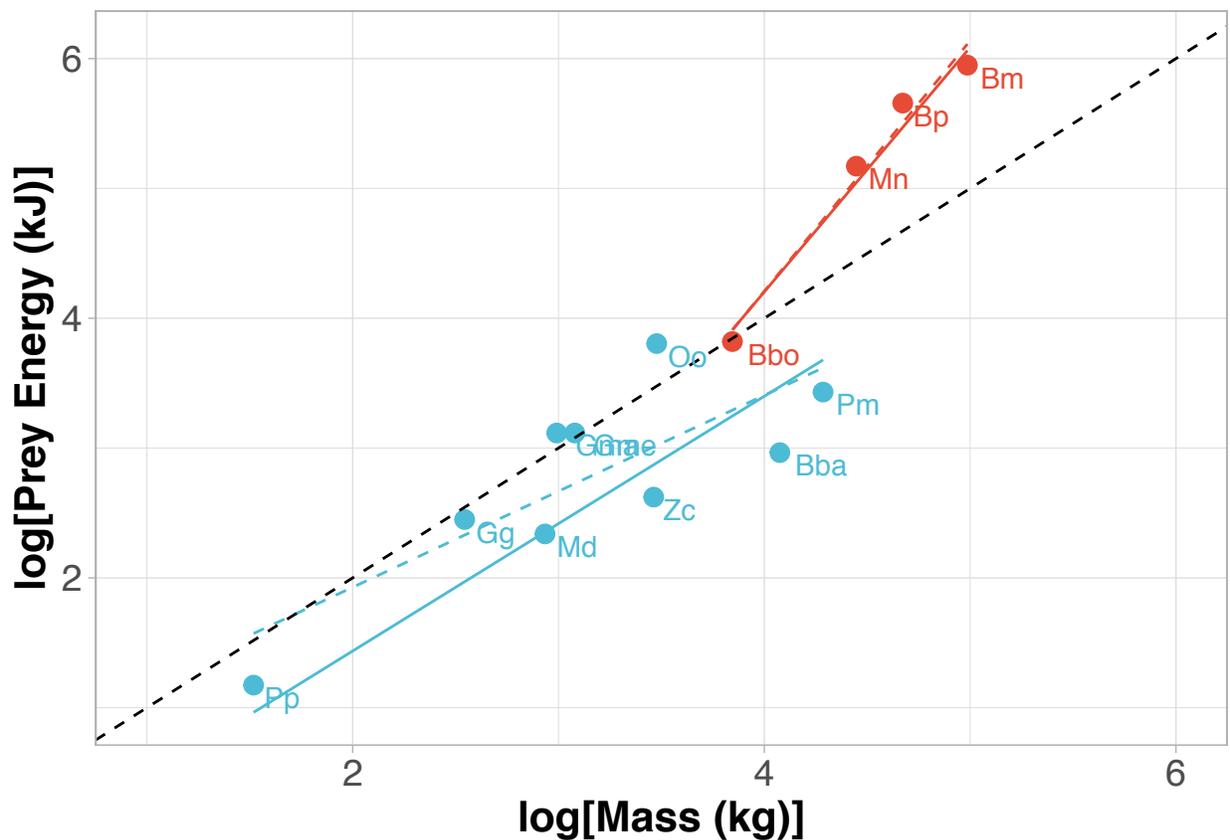
```

linetype = 2) +
geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
guides(size = FALSE, color = FALSE) +
theme_light() + theme(legend.position = "top") +
theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +

xlim(1,6) +
labs(x = "log[Mass (kg)]", y = "log[Prey Energy (kJ)]") +
geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
          aes(label = abbreviation, hjust = -0.3, vjust = 1.1) +
geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
          aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

```

fig_3



10.1 Construct output table

```

df.out <- smydata[,c("species", "fm", "x_mean", "y_mean")]
df.out$fitted_ols <- fitted(m.ols)
df.out$fitted_pgls <- fitted(m.pgls.nlme)
rownames(df.out) <- NULL
kable(df.out,
      caption = "Model outputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
kable_styling(latex_options = c("scale_down", "hold_position"))

```

Table 6: Model outputs

species	fm	x_mean	y_mean	fitted_ols	fitted_pgl
Balaenoptera_bonaerensis	Filter	3.8451	3.8204	3.9095	3.9114
Balaenoptera_musculus	Filter	4.9868	5.9494	6.1122	6.0619
Balaenoptera_physalus	Filter	4.6725	5.6574	5.5059	5.4700
Berardius_bairdii	Single-prey	4.0755	2.9647	3.4627	3.4731
Globicephala_macrorhynchus	Single-prey	2.9912	3.1162	2.6609	2.4093
Globicephala_melas	Single-prey	3.0792	3.1162	2.7259	2.4956
Grampus_griseus	Single-prey	2.5441	2.4496	2.3302	1.9706
Megaptera_novaeangliae	Filter	4.4472	5.1716	5.0711	5.0454
Mesoplodon_densirostris	Single-prey	2.9345	2.3385	2.6189	2.3536
Orcinus_orca	Single-prey	3.4771	3.8049	3.0202	2.8860
Phocoena_phocoena	Single-prey	1.5185	1.1740	1.5717	0.9645
Physeter_macrocephalus	Single-prey	4.2856	3.4314	3.6181	3.6791
Ziphius_cavirostris	Single-prey	3.4624	2.6223	3.0093	2.8715

pGLS for scaling of foraging capacity in cetaceans - **Figure 4** energetic efficiency vs body mass

Sep 01, 2019

Contents

1	Background	2
2	Load libraries	2
2.1	References	2
3	Read in tree data	3
4	Read in and explore trait data	3
4.1	Read in data and compute weighted means of energetic efficiencies (using frequency of occurrence of prey)	3
4.2	Rearrange and store them in a new data frame	3
5	Run model for MR = .45	5
5.1	Prepare data	5
5.2	Plot the data	6
5.3	Run OLS with feeding mode as a categorical predictor	7
5.4	Run pGLS with feeding mode as a categorical predictor	11
5.5	Estimate confidence intervals by bootstrapping	18
5.6	Extract summary statistics	32
5.7	Plot best models (OLS - dashed, PGLS - solid)	35
5.8	Quick clean up	36
6	Run model for MR = .61	37
6.1	Prepare data	37
6.2	Plot the data	39
6.3	Run OLS with feeding mode as a categorical predictor	40
6.4	Run a pGLS with feeding mode as a categorical predictor	44
6.5	Estimate confidence intervals by bootstrapping	51
6.6	Extract summary statistics	65
6.7	Plot best models (OLS - dashed, PGLS - solid)	68
6.8	Quick clean up	69
7	Run model for MR = .68	70
7.1	Prepare data	70
7.2	Plot the data	72
7.3	Run OLS with feeding mode as a categorical predictor	73
7.4	Run a pGLS with feeding mode as a categorical predictor	77
7.5	Estimate confidence intervals by bootstrapping	84
7.6	Extract summary statistics	98
7.7	Plot best models (OLS - dashed, PGLS - solid)	101
7.8	Quick clean up	102
8	Run model for MR = .75	103
8.1	Prepare data	103
8.2	Plot the data	105

8.3	Run OLS with feeding mode as a categorical predictor	106
8.4	Run a pGLS with feeding mode as a categorical predictor	110
8.5	Estimate confidence intervals by bootstrapping	117
8.6	Extract summary statistics	131
8.7	Plot best models (OLS - dashed, PGLS - solid)	134
8.8	Quick clean up	135
9	Combine best models	136
9.1	Combine parameters into a single data frame	136
9.2	Plot all models (OLS - thin lines, pGLS - thick lines)	137

1 Background

This is an R Markdown file documenting the PGLS analysis of scaling relationships in dive and foraging performance data from the two groups of cetaceans: Odontocetes and Mysticetes (Rorquals).

2 Load libraries

```
library(AICcmodavg) # Mazerolle 2017
library(ape) # Paradis and Schliep 2018
library(nlme) # Pinheiro et al. 2018
library(phytools) # Revell 2012
library(geiger) # Harmon et al. 2008
library(dplyr) # Wickham et al. 2018
library(ggplot2) # Wickham 2016
library(lme4) # Bates et al. 2015
library(rptR) # Stoffel et al. 2017
library(knitr) # Xie 2014, 2015, 2018
library(car) # Fox and Weisberg 2011
library(tinytex) # Xie 2019
library(kableExtra) # Zhu 2019
```

2.1 References

- Bates, D., Maechler, M., Bolker, B. and Walker, S. (2015) Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1), 1-48. doi:10.18637/jss.v067.i01.
- Fox, J. and Weisberg, S. (2011) *An {R} Companion to Applied Regression*, Second Edition. Thousand Oaks CA: Sage. URL: <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion>
- Harmon, L.J., Weir, J.T., Brock, C.D., Glor, R.E. and Challenger, W. (2008) GEIGER: investigating evolutionary radiations. *Bioinformatics* 24:129-131.
- Mazerolle, M.J. (2017) AICcmodavg: Model selection and multimodel inference based on (Q)AIC(c). R package version 2.1-1. <https://cran.r-project.org/package=AICcmodavg>.
- Paradis, E. and Schliep, K. (2018) ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. *Bioinformatics* 35: 526-528.
- Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D. and R Core Team (2018) nlme: Linear and Nonlinear Mixed Effects Models. R package version 3.1-137, <URL: <https://CRAN.R-project.org/package=nlme>>.
- Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.* 3 217-223. doi:10.1111/j.2041-210X.2011.00169.x
- Stoffel, M. A., Nakagawa, S. and Schielzeth, H. (2017) rptR: repeatability estimation and variance decomposition by generalized linear mixed-effects models. *Methods Ecol Evol*, 8: 1639-1644. doi:10.1111/2041-210X.12797
- Wickham, H. (2016) ggplot2: Elegant graphics for data analysis. Springer-Verlag New York
- Wickham, H., Romain, F., Lionel, H. and Müller, K. (2018) dplyr: A grammar of data manipulation. R package version 0.7.8. <https://CRAN.R-project.org/package=dplyr>
- Xie, Y. (2014) knitr: A comprehensive tool for reproducible research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595
- Xie, Y. (2015) *Dynamic documents with R and knitr*. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963
- Xie, Y. (2018) knitr: A general-purpose package for dynamic report generation in R. R package version 1.21.
- Xie, Y. (2019) tinytex: Helper Functions to Install and Maintain ‘TeX Live’, and Compile ‘LaTeX’ Documents. R package version 0.10. <https://CRAN.R-project.org/package=tinytex>
- Zhu, H. (2019) kableExtra: Construct Complex Table with ‘kable’ and Pipe Syntax. R package version 1.0.1. <https://CRAN.R-project.org/package=kableExtra>

3 Read in tree data

We used phylogenetic data produced by McGowen et al. (2009), downloaded as a Nexus file from TreeBase. The tree was edited in Mesquite to remove species without data on foraging rates, and to add timing of diversification of branches based on data published in McGowen et al. (2009).

```
mytree <- read.nexus("S10190_foragingsubset_Nov2018_edited.nex")
# plot(mytree)
```

4 Read in and explore trait data

4.1 Read in data and compute weighted means of energetic efficiencies (using frequency of occurrence of prey)

```
d_full <- read.csv("Cetacea model output v10.13 Zc fix_BOUT_EXTANT.csv")
d_full <- filter(d_full, Family != "Balaenidae")

d_full$MR.exponent = as.factor(d_full$MR.exponent)
d_full$M..kg. <- as.numeric(d_full$M..kg.)
d_full$Prey.W..g. <- as.numeric(d_full$Prey.W..g.)
d_full$Group <- ifelse(d_full$Family == "Balaenopteridae", "Rorqual",
                      ifelse(d_full$Family == "Balaenidae", "Balaenid", "Odontocete"))
d_full$Spec <- paste(d_full$Genus, d_full$Species, sep="_")
d_full$Spec <- as.factor(d_full$Spec)

abb <- character(nrow(d_full))
for (i in seq(1,nrow(d_full))) {
  if (d_full$Genus[i] == "Globicephala" || d_full$Genus[i] == "Berardius" ||
      d_full$Species[i] == "bonaerensis") {
    abb[i] <- paste(substr(d_full$Genus[i],1,1), substr(d_full$Species[i],1,2), sep = "")
  } else {
    abb[i] <- paste(substr(d_full$Genus[i],1,1), substr(d_full$Species[i],1,1), sep = "")
  }
}

d_full$abbreviation <- abb

d_full$x <- log10(d_full$M..kg.)
d_full$y <- log10(d_full$E_divesurf_max)

d_full.means <- d_full %>%
  group_by(Spec,MR.exponent) %>%
  summarize(wgtMean = weighted.mean(log10(E_divesurf_max),Percent))
spec <- unique(d_full$Spec)
```

4.2 Rearrange and store them in a new data frame

```
x_mean <- tapply(d_full$x, d_full$Spec, mean)

gr <- tapply(d_full$Family, d_full$Spec, unique)
tx <- tapply(d_full$Group, d_full$Spec, unique)
fm <- tx
```

Table 1: Dataset

Spec	gr	x_mean	fm	Group	abbreviation	wgtMean.45	wgtMean.61	wgtMean.68	wgtMean.75
Balaenoptera_bonaerensis	1	3.8451	Filter	Rorqual	Bbo	1.3289	1.2293	1.1386	1.0061
Balaenoptera_musculus	1	4.9868	Filter	Rorqual	Bm	2.0165	1.9307	1.8293	1.6606
Balaenoptera_physalus	1	4.6725	Filter	Rorqual	Bp	2.2749	2.1286	2.0246	1.8348
Berardius_bairdii	5	4.0755	Single-prey	Odontocete	Bba	0.7242	-0.3180	-0.0448	-0.3180
Globicephala_macrorhynchus	2	2.9912	Single-prey	Odontocete	Gma	1.0519	0.7918	0.6401	0.4703
Globicephala_melas	2	3.0792	Single-prey	Odontocete	Gme	1.2829	0.8406	0.6335	0.4257
Grampus_griseus	2	2.5441	Single-prey	Odontocete	Gg	1.1941	0.8068	0.6328	0.4577
Megaptera_novaeangliae	1	4.4472	Filter	Rorqual	Mn	2.1839	2.0144	1.8621	1.6563
Mesoplodon_densirostris	5	2.9345	Single-prey	Odontocete	Md	0.4639	0.0020	-0.2024	-0.4092
Orcinus_orca	2	3.4771	Single-prey	Odontocete	Oo	0.7544	0.6359	0.5393	0.4096
Phocoena_phocoena	3	1.5185	Single-prey	Odontocete	Pp	1.0350	0.8313	0.7372	0.6407
Physeter_macrocephalus	4	4.2856	Single-prey	Odontocete	Pm	0.7403	0.3781	0.1685	-0.0732
Ziphius_cavirostris	5	3.4624	Single-prey	Odontocete	Zc	0.4286	-0.0079	-0.2229	-0.4496

```

fm[tx=="Rorqual"] <- "Filter"
fm[tx=="Odontocete"] <- "Single-prey"
abbreviation <- tapply(d_full$abbreviation, d_full$Spec, unique)

data.spec <- cbind(gr, x_mean)
df.spec <- data.frame(Spec = row.names(data.spec), data.spec, row.names =
                      rownames(data.spec), check.rows = TRUE, check.names = TRUE)
df.spec$gr <- factor(df.spec$gr)
df.spec$fm <- factor(fm)
df.spec$Group <- factor(tx)
df.spec$abbreviation <- factor(abbreviation)

df.spec2 <- df.spec %>% left_join(subset(data.frame(filter(d_full.means,
                                                         MR.exponent == 0.45)),
                                   select = c(Spec,wgtMean)), by = "Spec")
colnames(df.spec2)[ncol(df.spec2)] <- paste(colnames(df.spec2)[ncol(df.spec2)], ".45",
                                             sep = "")
df.spec2 <- df.spec2 %>% left_join(subset(data.frame(filter(d_full.means,
                                                         MR.exponent == 0.61)),
                                   select = c(Spec,wgtMean)), by = "Spec")
colnames(df.spec2)[ncol(df.spec2)] <- paste(colnames(df.spec2)[ncol(df.spec2)], ".61",
                                             sep = "")
df.spec2 <- df.spec2 %>% left_join(subset(data.frame(filter(d_full.means,
                                                         MR.exponent == 0.68)),
                                   select = c(Spec,wgtMean)), by = "Spec")
colnames(df.spec2)[ncol(df.spec2)] <- paste(colnames(df.spec2)[ncol(df.spec2)], ".68",
                                             sep = "")
df.spec2 <- df.spec2 %>% left_join(subset(data.frame(filter(d_full.means,
                                                         MR.exponent == 0.75)),
                                   select = c(Spec,wgtMean)), by = "Spec")
colnames(df.spec2)[ncol(df.spec2)] <- paste(colnames(df.spec2)[ncol(df.spec2)], ".75",
                                             sep = "")

df.spec <- df.spec2

kable(df.spec2,
      caption = "Dataset",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")

```

```
row.names(df.spec) <- df.spec$Spec
```

4.2.1 Rearrange the row order in df.spec to match mytree

```
df.spec <- df.spec[match(mytree$tip.label, rownames(df.spec)),]
```

5 Run model for MR = .45

5.1 Prepare data

5.1.1 Get rid of rows with NAs - subset the data

```
smydata <- df.spec  
smydata$y_mean <- smydata$wgtMean.45  
smydata <- smydata[!is.na(smydata$y_mean),]  
smydata <- smydata[!is.na(smydata$x_mean),]  
smydata$fm <- factor(smydata$fm)  
smydata$Group <- smydata$Group  
colnames(smydata)[1] <- "species"
```

5.1.2 Adjust tree - drop species for which data are missing

```
smytree <- drop.tip(mytree, mytree$tip.label[-match(smydata$species, mytree$tip.label)])  
plot(smytree)
```

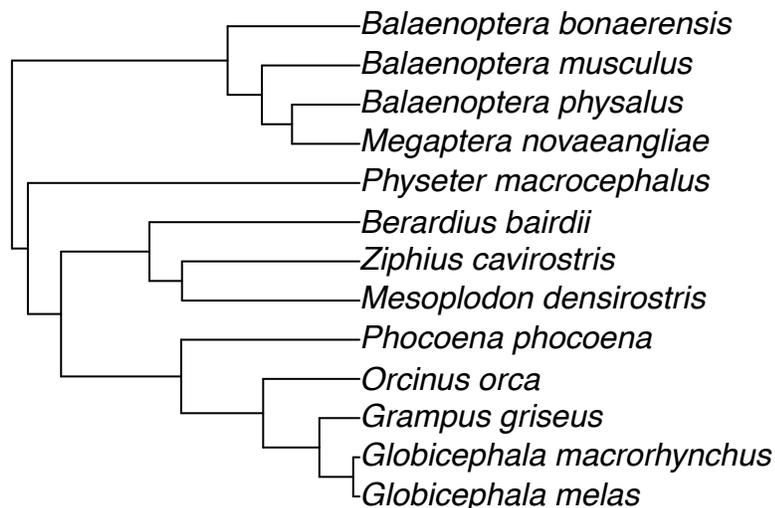


Table 2: Model inputs

species	gr	x_mean	fm	Group	abbreviation	wgtMean.45	wgtMean.61	wgtMean.68	wgtMean.75	y_mean
Balaenoptera_bonaerensis	1	3.8451	Filter	Rorqual	Bbo	1.3289	1.2293	1.1386	1.0061	1.3289
Balaenoptera_musculus	1	4.9868	Filter	Rorqual	Bm	2.0165	1.9307	1.8293	1.6606	2.0165
Balaenoptera_physalus	1	4.6725	Filter	Rorqual	Bp	2.2749	2.1286	2.0246	1.8348	2.2749
Berardius_bairdii	5	4.0755	Single-prey	Odontocete	Bba	0.7242	-0.3180	-0.0448	-0.3180	0.7242
Globicephala_macrorhynchus	2	2.9912	Single-prey	Odontocete	Gma	1.0519	0.7918	0.6401	0.4703	1.0519
Globicephala_melas	2	3.0792	Single-prey	Odontocete	Gme	1.2829	0.8406	0.6335	0.4257	1.2829
Grampus_griseus	2	2.5441	Single-prey	Odontocete	Gg	1.1941	0.8068	0.6328	0.4577	1.1941
Megaptera_novaeangliae	1	4.4472	Filter	Rorqual	Mn	2.1839	2.0144	1.8621	1.6563	2.1839
Mesoplodon_densirostris	5	2.9345	Single-prey	Odontocete	Md	0.4639	0.0020	-0.2024	-0.4092	0.4639
Orcinus_orca	2	3.4771	Single-prey	Odontocete	Oo	0.7544	0.6359	0.5393	0.4096	0.7544
Phocoena_phocoena	3	1.5185	Single-prey	Odontocete	Pp	1.0350	0.8313	0.7372	0.6407	1.0350
Physeter_macrocephalus	4	4.2856	Single-prey	Odontocete	Pm	0.7403	0.3781	0.1685	-0.0732	0.7403
Ziphius_cavirostris	5	3.4624	Single-prey	Odontocete	Zc	0.4286	-0.0079	-0.2229	-0.4496	0.4286

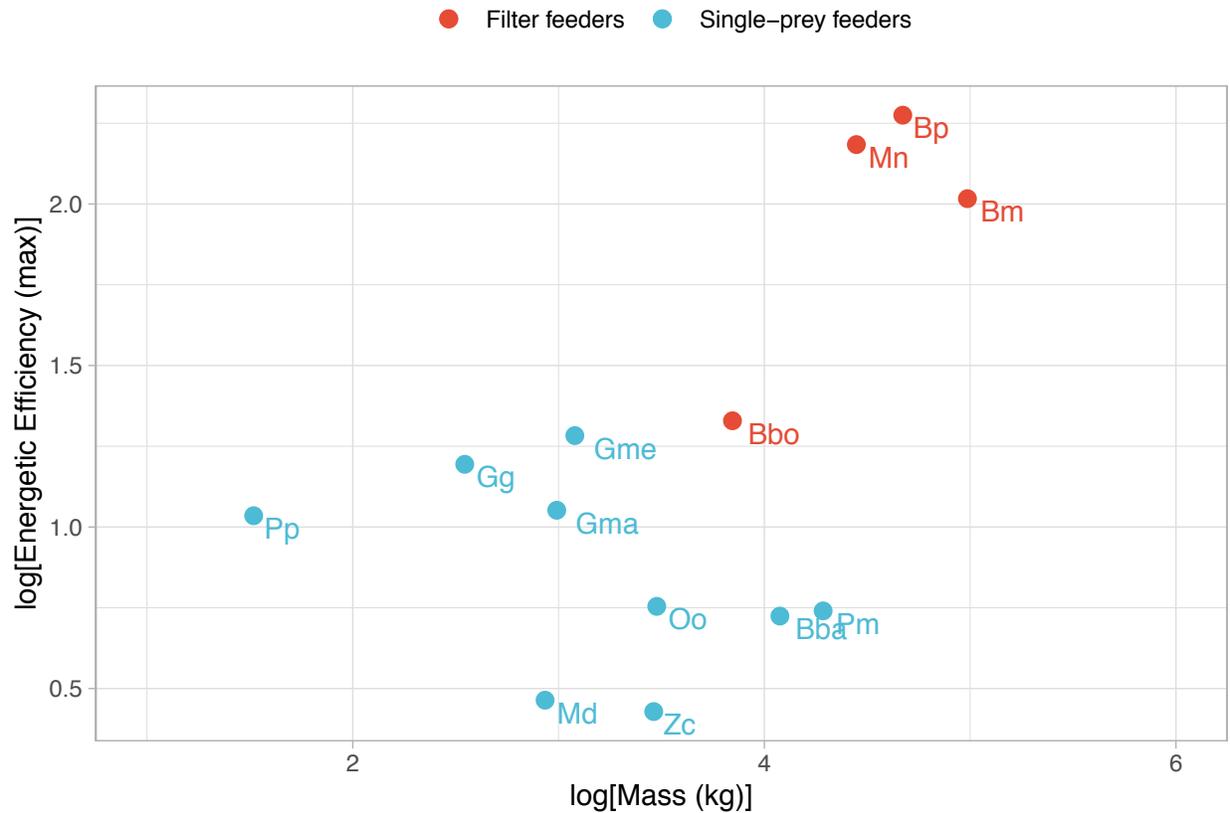
5.1.3 Rearrange the row order in smydata to match smytree

```
smydata <- smydata[match(smytree$tip.label, rownames(smydata)),]
rownames(smydata) <- NULL
kable(smydata,
      caption = "Model inputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")
```

```
rownames(smydata) <- smydata$species
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_45_smydata.rds")
saveRDS(smydata,fil)
```

5.2 Plot the data

```
ggplot(smydata, aes(x_mean, y = value, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), shape = 16, size = 3,
            aes(y = y_mean, color = "#4DBBD5FF")) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), shape = 16, size = 3,
            aes(y = y_mean, color = "#E64B35FF")) +
  scale_color_manual(name = "",
                    values = c("#E64B35FF", "#4DBBD5FF"),
                    labels = c("Filter feeders", "Single-prey feeders")) +
  theme_light() + theme(legend.position = "top") +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1)
```



5.3 Run OLS with feeding mode as a categorical predictor

5.3.1 Run OLS and model reduction using ML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "ML")
summary(m.ols)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 10.87598 13.70072 -0.437989
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.1545966 1.6228922 -0.7114438 0.4948
## fmSingle-prey  2.5237039 1.6758760 1.5059013 0.1664
## x_mean         0.6920041 0.3600599 1.9219139 0.0868
## fmSingle-prey:x_mean -0.8558047 0.3823919 -2.2380305 0.0520
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996 0.964
## fmSingle-prey:x_mean 0.938 -0.989 -0.942
##
```

```
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.69631483 -0.70863061  0.09076137  0.78355484  1.67088878
##
## Residual standard error: 0.250262
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.ols)
```

```
## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)    1 203.73959 <.0001
## fm              1  36.91984 0.0002
## x_mean         1   0.30318 0.5953
## fm:x_mean      1   5.00878 0.0520
```

```
m.ols.2 <- update(m.ols, ~ . - fm:x_mean)
anova(m.ols, m.ols.2)
```

```
##      Model df      AIC      BIC  logLik  Test L.Ratio p-value
## m.ols      1  5 10.87598 13.70073 -0.437989
## m.ols.2    2  4 14.62796 16.88775 -3.313977 1 vs 2 5.751977 0.0165
```

5.3.1.1 Compare to an intercept-only model

```
m.ols.0 <- gls(y_mean ~ 1, data = smydata, method = "ML")
anova(m.ols, m.ols.0)
```

```
##      Model df      AIC      BIC  logLik  Test L.Ratio p-value
## m.ols      1  5 10.87598 13.70073 -0.437989
## m.ols.0    2  2 27.48475 28.61464 -11.742372 1 vs 2 22.60877 <.0001
```

```
m.ols.p <- anova(m.ols, m.ols.0)$`p-value`[2]
```

5.3.2 Estimate final model using REML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
summary(m.ols)
```

```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC  logLik
## 18.83645 19.82257 -4.418226
##
## Coefficients:
##              Value Std.Error  t-value p-value
## (Intercept)  -1.1545966  1.6228922 -0.7114438  0.4948
## fmSingle-prey  2.5237039  1.6758760  1.5059013  0.1664
## x_mean        0.6920041  0.3600599  1.9219139  0.0868
## fmSingle-prey:x_mean -0.8558047  0.3823919 -2.2380305  0.0520
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean        -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
```

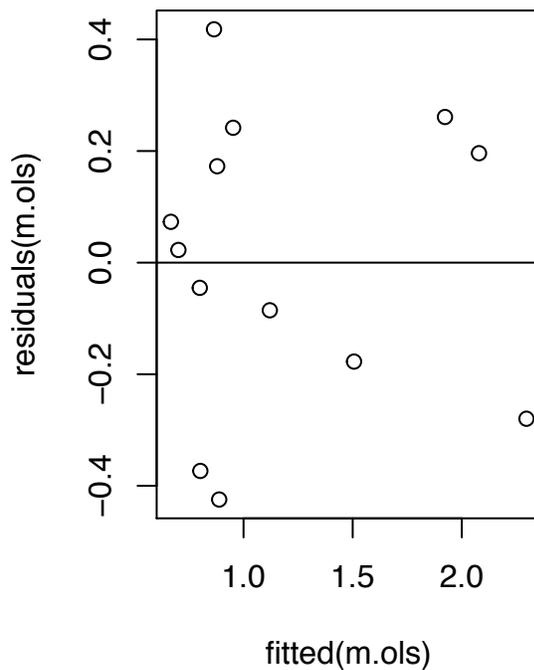
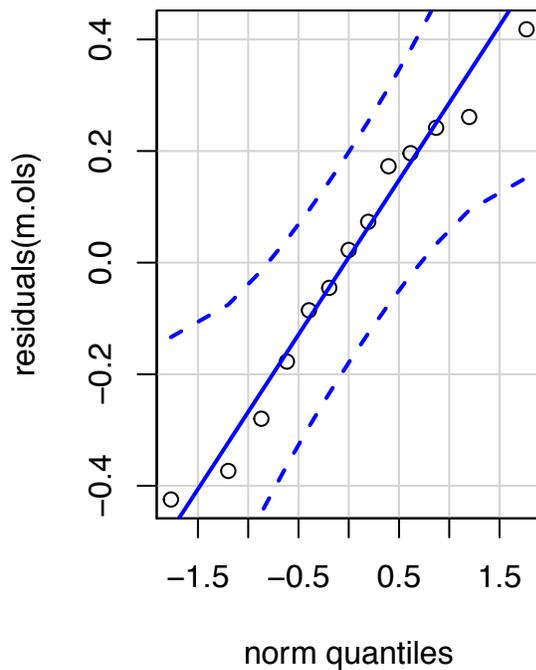
```
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.41141925 -0.58961630 0.07551802 0.65195704 1.39026350
##
## Residual standard error: 0.3007774
## Degrees of freedom: 13 total; 9 residual

m.ols.param <- as.data.frame(t(summary(m.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.ols.param <- m.ols.param[5:8]
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_45_m_ols_param.rds")
saveRDS(m.ols.param,fil)
```

5.3.2.1 Model diagnostics

5.3.2.1.1 QQ-plot and Residuals vs fitted plot

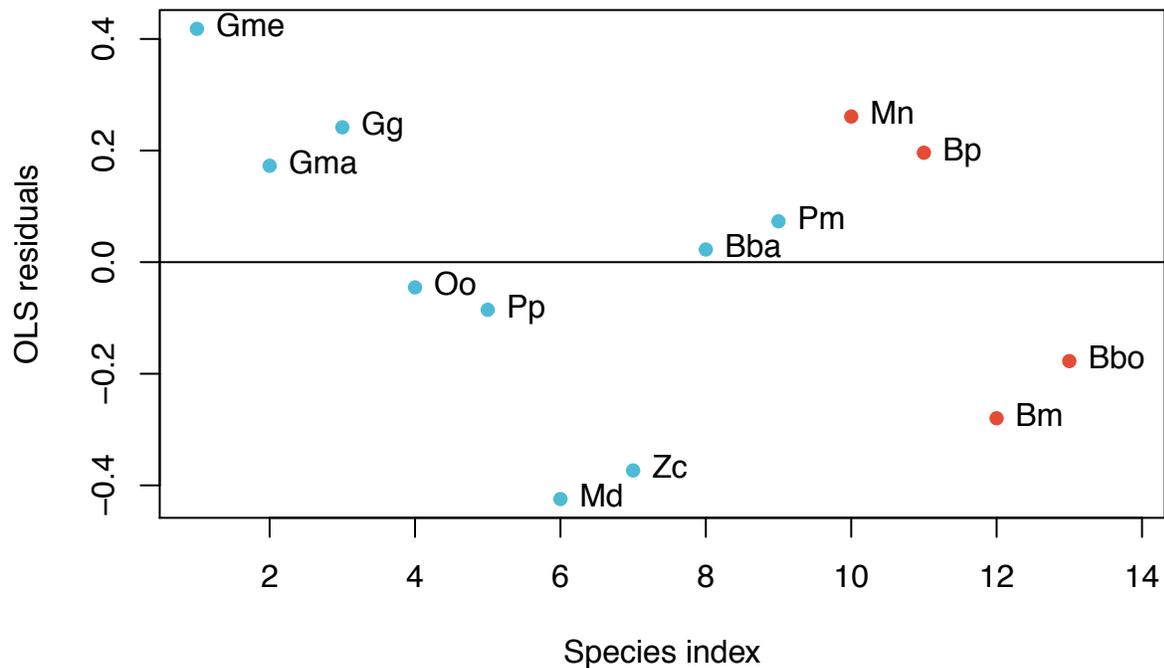
```
par(mfrow=c(1,2))
qqPlot(residuals(m.ols), id=FALSE)
plot(fitted(m.ols), residuals(m.ols))
abline(0,0)
```



5.3.3 Evaluate for phylogenetic correlation

5.3.3.1 Plot residuals ordered “by phylogeny” (i.e. in the order of tips of the phylogenetic tree)

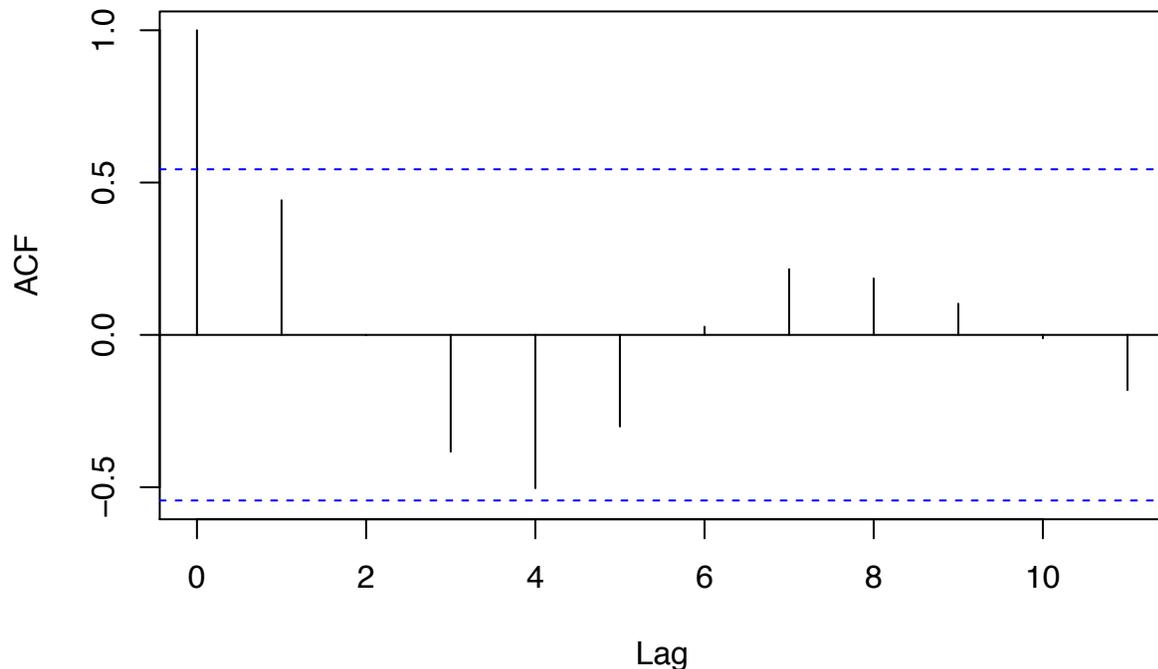
```
is_tip <- smytree$edge[,2]<=length(smytree$tip.label)
ordered_tips <- smytree$edge[is_tip,2] # extract the order of tree tips
oj <- residuals(m.ols)
tl <- smytree$tip.label[ordered_tips]
res <- oj[tl]
plot(oj[tl], pch=16, ylab="OLS residuals", xlab="Species index",
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata[tl,"fm"])],
     xlim=c(1,13.8))
abline(0,0)
text(oj[tl], labels=abbreviation[tl], pos=4)
```



5.3.3.2 Plot autocorrelation function of residuals ordered “by phylogeny”

```
acf(res, main="Series: residuals sorted by phylogeny")
```

Series: residuals sorted by phylogeny



5.4 Run pGLS with feeding mode as a categorical predictor

5.4.1 Estimate Pagel's λ (amount of phylogenetic signal) for each trait separately

Can be informative, but only λ for the entire model should be considered when deciding on whether running pGLS is appropriate.

```
lambdax <- phylosig(smytree, smydata$x_mean, method = "lambda", test = T)
```

```
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
```

```
lambday <- phylosig(smytree, smydata$y_mean, method = "lambda", test = T)
```

```
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
```

```
cbind(lambdax, lambday)
```

```
##      lambdax      lambday
## lambda 1.014327  0.9554531
## logL   -12.69024  -6.180711
## logL0  -17.41681 -11.74237
## P      0.002107892 0.0008524896
```

5.4.2 Plot likelihood surface for Pagel's λ for model without feeding mode as a covariate

λ estimate for the model marked in red.

```
lambda <- seq(0, 1, length.out = 500)
```

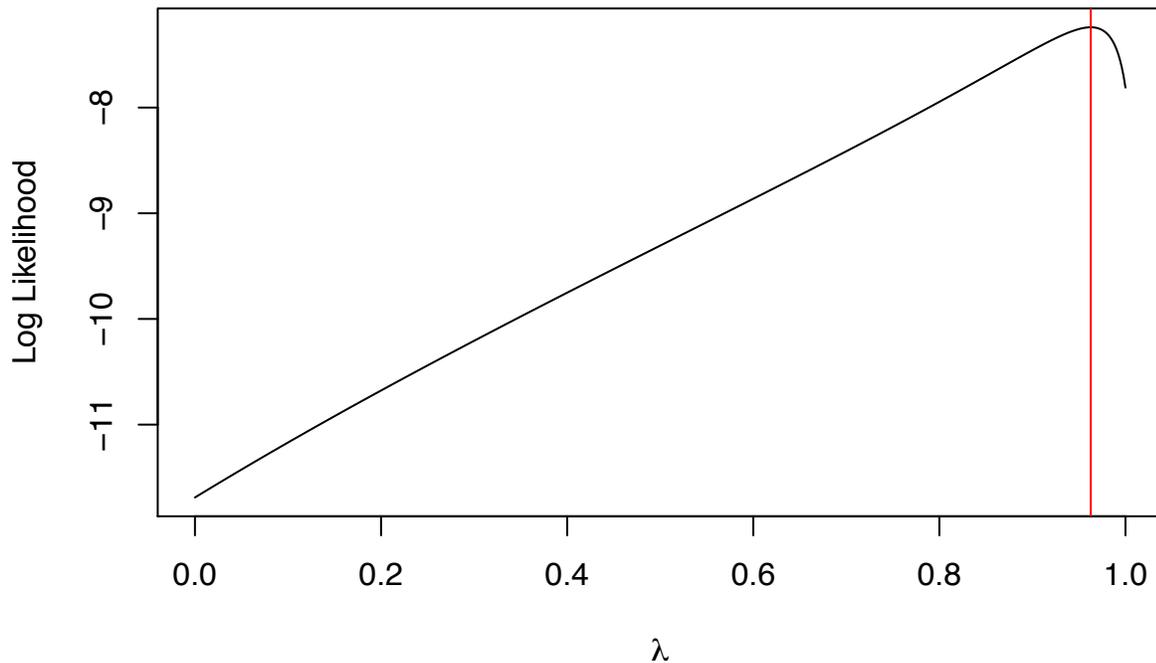
```
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ x_mean, smydata,
```

```

method = "REML", correlation = corPagel(value = lambda, phy = smytree,
                                       fixed = TRUE)))
plot(lik ~ lambda, type = "l", main =
     expression(paste("Prey energy to body mass Likelihood Plot for ", lambda)),
     ylab = "Log Likelihood", xlab = expression(lambda))
m.pa.only <- gls(y_mean ~ x_mean, data = smydata, correlation =
                corPagel(value = 0, phy = smytree, fixed = FALSE), method = "REML")
abline(v = m.pa.only$modelStruct[1], col = "red")

```

Prey energy to body mass Likelihood Plot for λ



5.4.3 Estimate Pagel's λ using REML

If λ is estimated to be greater than 1, fix it at 1, if smaller than 0, fix it at 0.

$\lambda = 0$ suggests that the relationship between predictor and response variables is unrelated to phylogeny, while $\lambda = 1$ indicates that traits have evolved under Brownian motion on the given phylogeny. Intermediate values of λ indicate that traits have evolved according to a process in which the effect of phylogeny is weaker than in the Brownian model, while values of $\lambda > 1$ can arise if, for instance, traits are more similar than predicted by Brownian motion, given the phylogeny.

```

m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
                 corPagel(1, phy = smytree, fixed = FALSE), method = "REML")
summary(m.pgls.nlme)

```

```

## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 18.09733 19.28068 -3.048664

```

```
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
##   lambda
## 0.8540079
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -0.8686509 1.3550709 -0.6410372  0.5375
## fmSingle-prey  1.9812393 1.4415123  1.3744172  0.2026
## x_mean        0.6203115 0.2981143  2.0807842  0.0672
## fmSingle-prey:x_mean -0.7247022 0.3273161 -2.2140745  0.0541
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.940
## x_mean        -0.975  0.917
## fmSingle-prey:x_mean  0.888 -0.965 -0.911
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -0.9466745 -0.5188338  0.2075485  0.6955675  1.3598168
##
## Residual standard error: 0.3616261
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.pgls.nlme)
```

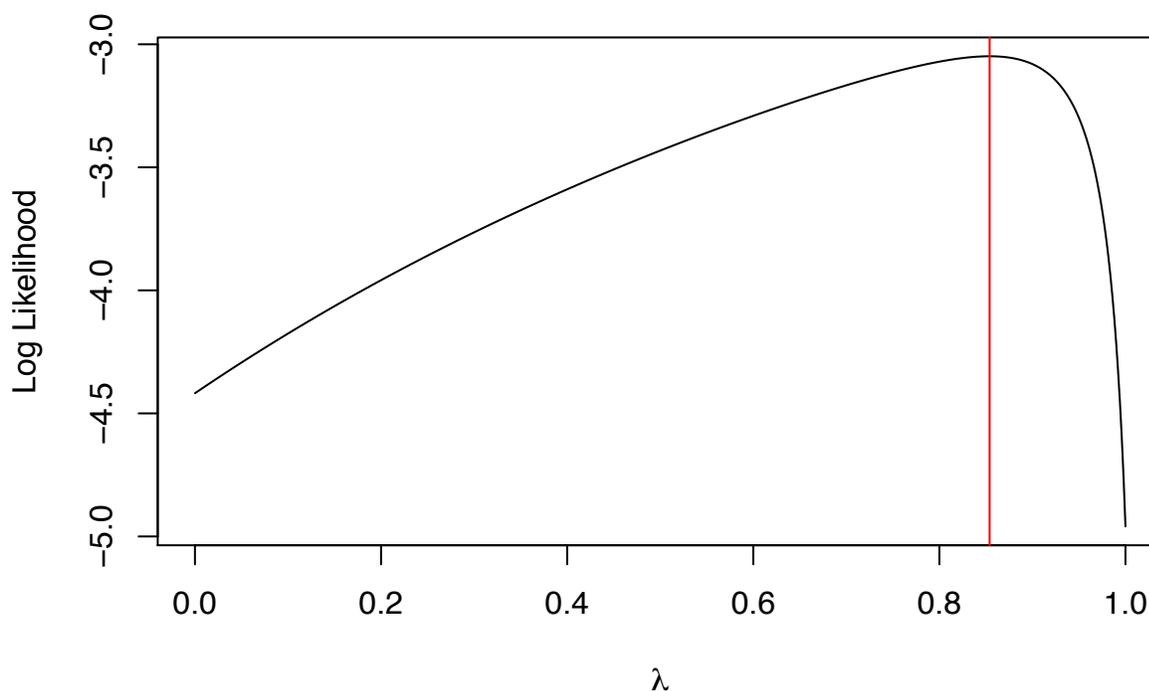
```
## Denom. DF: 9
##          numDF  F-value p-value
## (Intercept)    1 45.52625  0.0001
## fm              1  9.92307  0.0117
## x_mean         1  0.02421  0.8798
## fm:x_mean      1  4.90213  0.0541
```

```
lambda.est <- as.numeric(m.pgls.nlme$modelStruct[1])
if(lambda.est > 1){lambda.est <- 1} else if(lambda.est < 0){lambda.est <- 0}
```

5.4.4 Plot likelihood surface for Pagel's λ - our estimate marked in red

```
lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ fm * x_mean, smydata,
                                                method = "REML", correlation =
                                                corPagel(value = lambda, phy = smytree, fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
     expression(paste("Energetic Efficiency to Body mass Likelihood Plot for ", lambda)),
     ylab = "Log Likelihood", xlab = expression(lambda)),
abline(v = m.pgls.nlme$modelStruct, col = "red")
```

Energetic Efficiency to Body mass Likelihood Plot for λ



5.4.5 Run pGLS and model reduction with a fixed Pagel's λ (using ML)

```
m.pgls.nlmf <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")
summary(m.pgls.nlmf)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 10.52566 13.35041 -0.2628312
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
##   lambda
## 0.8540079
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -0.8686509 1.3550709  -0.6410372  0.5375
## fmSingle-prey  1.9812393 1.4415123  1.3744172  0.2026
## x_mean        0.6203115 0.2981143  2.0807842  0.0672
## fmSingle-prey:x_mean -0.7247022 0.3273161 -2.2140745  0.0541
##
## Correlation:
```

```

##          (Intr) fmSng- x_mean
## fmSingle-prey      -0.940
## x_mean            -0.975  0.917
## fmSingle-prey:x_mean  0.888 -0.965 -0.911
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.1377612 -0.6235606  0.2494422  0.8359682  1.6342964
##
## Residual standard error: 0.3008911
## Degrees of freedom: 13 total; 9 residual
anova(m.pgls.nlme)

## Denom. DF: 9
##          numDF  F-value p-value
## (Intercept)    1 45.52625  0.0001
## fm              1  9.92307  0.0117
## x_mean          1  0.02421  0.8798
## fm:x_mean       1  4.90213  0.0541

m.pgls.nlme.2 <- update(m.pgls.nlme, ~ . - fm:x_mean)
anova(m.pgls.nlme, m.pgls.nlme.2)

##          Model df      AIC      BIC      logLik  Test  L.Ratio
## m.pgls.nlme      1  5 10.52566 13.35041 -0.2628312
## m.pgls.nlme.2    2  4 14.17829 16.43808 -3.0891429 1 vs 2 5.652623
##          p-value
## m.pgls.nlme
## m.pgls.nlme.2  0.0174

m.pgls.fm <- gls(y_mean ~ fm, data = smydata, correlation =
                corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")
summary(m.pgls.fm)

## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm
## Data: smydata
##      AIC      BIC      logLik
## 12.2009 13.89575 -3.100452
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
## 0.8540079
##
## Coefficients:
##          Value Std.Error  t-value p-value
## (Intercept)  1.881468 0.3364174  5.592659  0.0002
## fmSingle-prey -1.118643 0.3995680 -2.799632  0.0173
##
## Correlation:
##          (Intr)
## fmSingle-prey -0.842
##

```

```
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.4763583 -0.1030755  0.3607803  0.8079411  1.3894927
##
## Residual standard error: 0.3742885
## Degrees of freedom: 13 total; 11 residual
```

```
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC      logLik
## 10.52566 13.35041 -0.2628312
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
##      lambda
## 0.8540079
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -0.8686509 1.3550709 -0.6410372  0.5375
## fmSingle-prey  1.9812393 1.4415123  1.3744172  0.2026
## x_mean        0.6203115 0.2981143  2.0807842  0.0672
## fmSingle-prey:x_mean -0.7247022 0.3273161 -2.2140745  0.0541
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.940
## x_mean        -0.975  0.917
## fmSingle-prey:x_mean  0.888 -0.965 -0.911
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.1377612 -0.6235606  0.2494422  0.8359682  1.6342964
##
## Residual standard error: 0.3008911
## Degrees of freedom: 13 total; 9 residual
```

5.4.5.1 Compare to an intercept-only model

```
m.pgls.nlme.0 <- gls(y_mean ~ 1, smydata, correlation = corPagel(value = lambda.est,
  phy = smytree, fixed = TRUE), method = "ML")
anova(m.pgls.nlme, m.pgls.nlme.0)
```

```
##           Model df      AIC      BIC   logLik   Test  L.Ratio p-value
## m.pgls.nlme     1  5 10.52566 13.35041 -0.262831
## m.pgls.nlme.0   2  2 17.19461 18.32451 -6.597306 1 vs 2 12.66895  0.0054
```

```
m.pgls.p <- anova(m.pgls.nlme, m.pgls.nlme.0)$`p-value`[2]
```

5.4.6 Estimate final model using REML

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "REML")
summary(m.pgls.nlme)
```

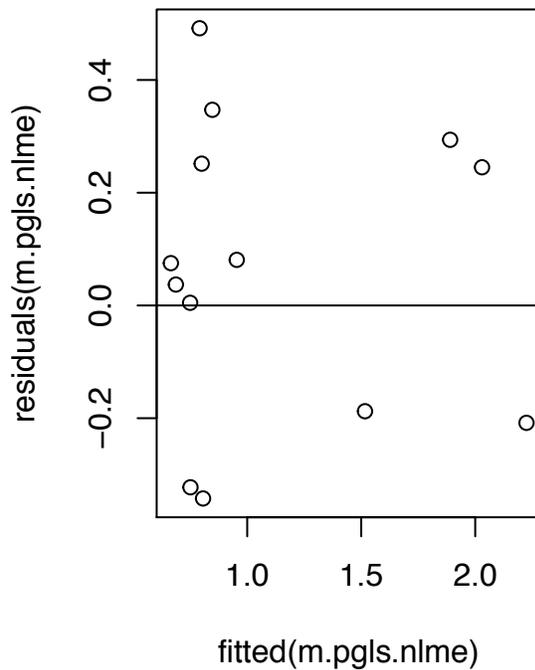
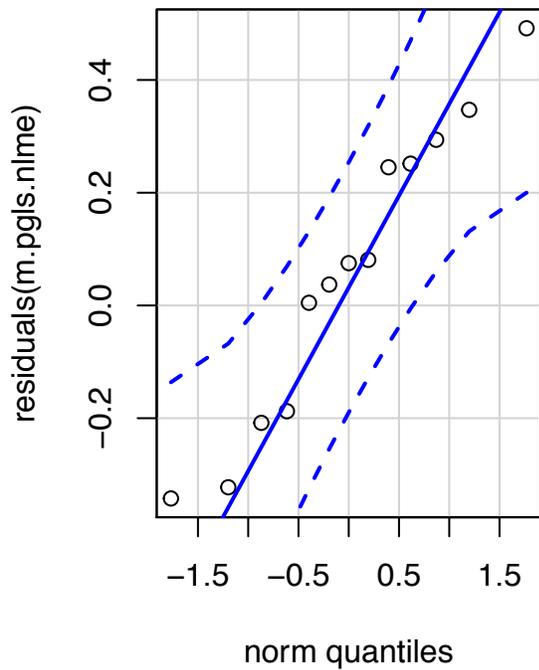
```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 16.09733 17.08345 -3.048664
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
##   lambda
## 0.8540079
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -0.8686509 1.3550709 -0.6410372  0.5375
## fmSingle-prey  1.9812393 1.4415123  1.3744172  0.2026
## x_mean         0.6203115 0.2981143  2.0807842  0.0672
## fmSingle-prey:x_mean -0.7247022 0.3273161 -2.2140745  0.0541
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey   -0.940
## x_mean          -0.975  0.917
## fmSingle-prey:x_mean 0.888 -0.965 -0.911
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -0.9466745 -0.5188338  0.2075485  0.6955675  1.3598168
##
## Residual standard error: 0.3616261
## Degrees of freedom: 13 total; 9 residual
```

```
m.pgls.param <- as.data.frame(t(summary(m.pgls.nlme)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.pgls.param <- m.pgls.param[5:8]
```

5.4.7 Model diagnostics

5.4.7.1 QQ-plot and Residuals vs fitted plot

```
par(mfrow = c(1,2))
qqPlot(residuals(m.pgls.nlme), id = FALSE)
plot(fitted(m.pgls.nlme), residuals(m.pgls.nlme))
abline(0,0)
```



5.5 Estimate confidence intervals by bootstrapping

5.5.1 Bootstrap and compute percentile confidence intervals

```
d_sub <- filter(d_full, MR.exponent==.45)
index <- d_sub %>% group_by(Spec) %>% summarize(ix=length(y))
index # number of prey categories for each species
```

```
## # A tibble: 13 x 2
##   Spec                ix
##   <fct>              <int>
## 1 Balaenoptera_bonaerensis    5
## 2 Balaenoptera_musculus      7
## 3 Balaenoptera_physalus      7
## 4 Berardius_bairdii         19
## 5 Globicephala_macrorhynchus 12
## 6 Globicephala_melas       12
## 7 Grampus_griseus           5
## 8 Megaptera_novaeangliae     8
## 9 Mesoplodon_densirostris    3
## 10 Orcinus_orca              12
## 11 Phocoena_phocoena         5
## 12 Physeter_macrocephalus    18
## 13 Ziphius_cavirostris      16
```

```

smydata.orig <- smydata
y_mean <- by(d_sub, d_sub$Spec, with, weighted.mean(y, Percent))
spec <- spec[match(spec,smydata$species)]

runpGls <- function(smydata,smytree){
  out <- tryCatch(
    {
      model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
        corPagel(lambda.est, phy = smytree, fixed = FALSE),
        method = "REML")
      as.numeric(model.pgls$modelStruct[1])
    },
    error=function(cond) {
      return(NA)
    }
  )
}

a.ols <- matrix(nrow=10000,ncol=4)
a.pgls <- matrix(nrow=10000,ncol=4)
b <- matrix(nrow=10000,ncol=length(spec))
boot.lambdas <- rep(NA,10000)
for(i in 1:10000){
  for(sp in 1:length(spec)){
    ix <- sample(1:index$ix[index$Spec==spec[sp]], replace = T)
    y_mean[sp] <- sum(d_sub[d_sub$Spec==spec[sp],"y"][ix]*
      d_sub[d_sub$Spec==spec[sp],"Percent"][ix])/
      sum(d_sub[d_sub$Spec==spec[sp],"Percent"][ix])
  }
  smydata$y_mean <- y_mean

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
  myout <- runpGls(smydata,smytree)
  boot.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(lambda.est, phy = smytree, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(l.est, phy = smytree, fixed = TRUE), method = "REML")
  }

  a.ols[i,] <- c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4])
  a.pgls[i,] <- c(coef(model.pgls)[1],coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],coef(model.pgls)[3]+coef(model.pgls)[4])
  b[i,] <- predict(model.ols)
}

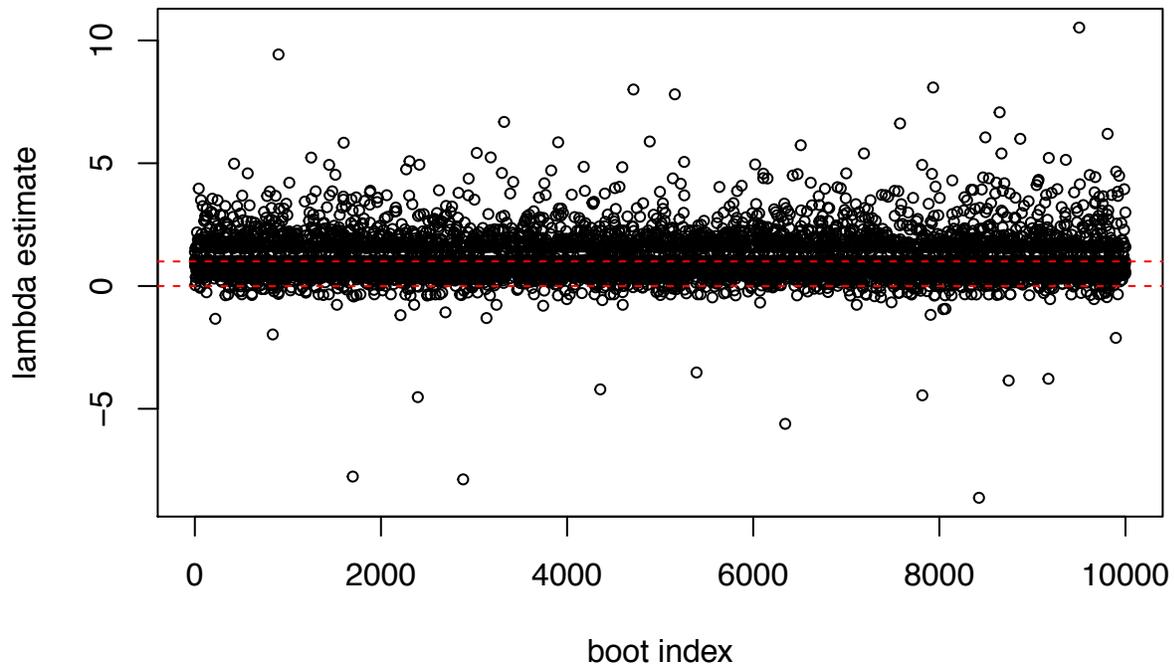
# number of pGLS models, where lambda could not be estimated ==> used original value:

```

```
sum(is.na(boot.lambdas))
```

```
## [1] 120
```

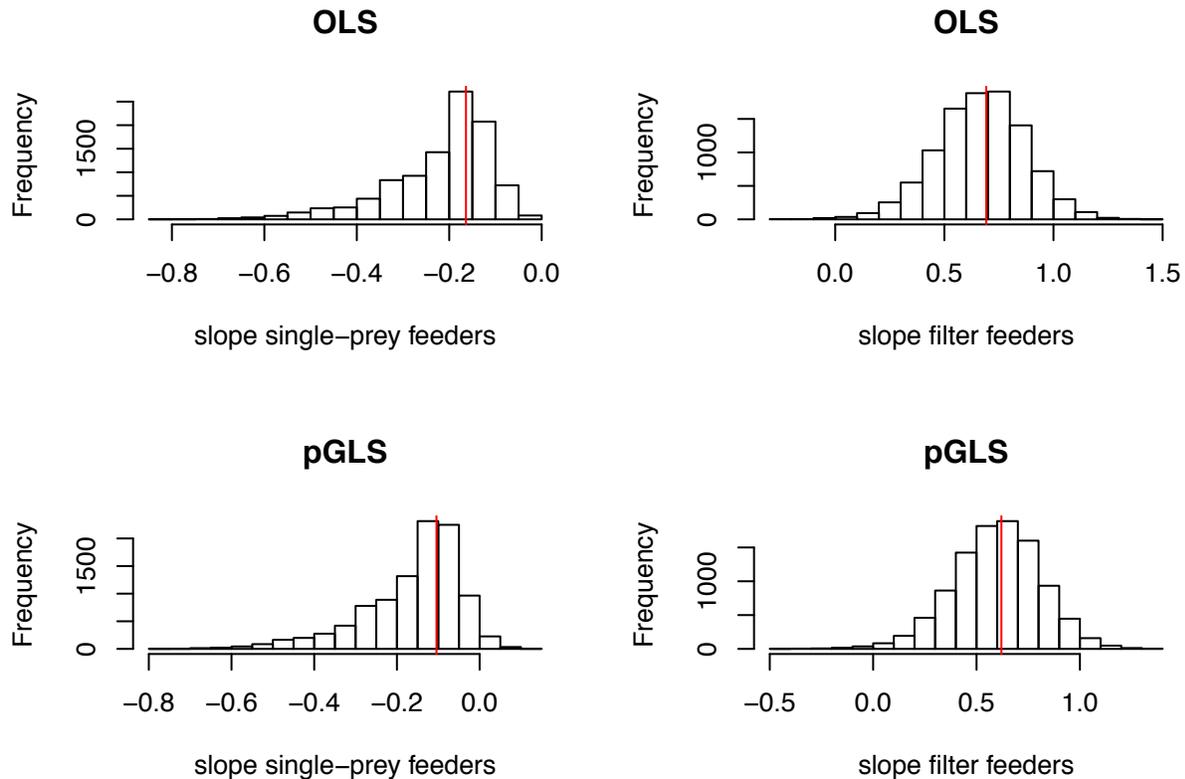
```
plot(boot.lambdas, cex=.7, xlab="boot index", ylab="lambda estimate")  
abline(h=0,lty="dashed",col="red")  
abline(h=1,lty="dashed",col="red")
```



```
preds <- apply(b, 2, quantile, c(0.025, 0.975))  
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",  
             "Figure4_45_bootstrap_b.rds")  
saveRDS(b,fil)  
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",  
             "Figure4_45_bootstrap_preds.rds")  
saveRDS(preds,fil)  
  
df.boot.ols <- data.frame(cbind(t(m.ols.param),t(t(apply(a.ols, 2, mean))),  
                             t(apply(a.ols, 2, quantile, c(0.025, 0.975)))))  
names(df.boot.ols) <- c("obs","bootest","lowerCI","upperCI")  
df.boot.pgls <- data.frame(cbind(t(m.pgls.param),t(t(apply(a.pgls, 2, mean))),  
                               t(apply(a.pgls, 2, quantile, c(0.025, 0.975)))))  
names(df.boot.pgls) <- c("obs","bootest","lowerCI","upperCI")  
  
par(mfrow=c(2,2))  
hist(a.ols[,4], xlab="slope single-prey feeders", main="OLS")  
abline(v=m.ols.param[4], col="red")  
hist(a.ols[,3], xlab="slope filter feeders", main="OLS")
```

```
abline(v=m.ols.param[3], col="red")

hist(a.pgls[,4], xlab="slope single-prey feeders", main="pGLS")
abline(v=m.pgls.param[4], col="red")
hist(a.pgls[,3], xlab="slope filter feeders", main="pGLS")
abline(v=m.pgls.param[3], col="red")
```



5.5.2 Compute BCa (bias-corrected and accelerated) confidence intervals

Corrects for bias and skewness in the distribution of bootstrap estimates.

Based on <https://blogs.sas.com/content/iml/2017/07/12/bootstrap-bca-interval.html>

```
smydata <- smydata.orig

# compute bias-correction factor from the proportion of bootstrap estimates
# that are less than the observed estimate

bootBC <- function(bootEst, Est){
  B <- ncol(bootEst)*nrow(bootEst) # number of bootstrap samples
  propLess <- sum(bootEst < Est)/B # proportion of replicates less than observed stat
  z0 <- qnorm(propLess) # bias correction
  return(z0)
}

z0.ols <- numeric()
for (i in 1:ncol(a.ols)){
```

```

z0.ols[i] <- bootBC(t(t(a.ols[,i])),as.numeric(m.ols.param[i]))
}

z0.pgls <- numeric()
for (i in 1:ncol(a.pgls)){
z0.pgls[i] <- bootBC(t(t(a.pgls[,i])),as.numeric(m.pgls.param[i]))
}

# compute acceleration factor, which is related to the skewness of bootstrap estimates.
# Use jackknife replicates to estimate.

jStat.ols <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jStat.pgls <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jack.lambdas <- rep(NA,nrow(smydata))
for (i in 1:nrow(smydata)) {
  d_sub <- subset(d_full, Spec==smydata$species[i] & MR.exponent==.45)
  y_mean.j <- numeric()
  for(j in 1:nrow(d_sub)){
    d_sub.j <- d_sub[-j,]
    y_mean.j[j] <- sum(d_sub.j$y*d_sub.j$Percent)/sum(d_sub.j$Percent)
  }
  smydata.j <- smydata
  smydata.j$y_mean[i] <- mean(y_mean.j)
  pruned.tree <- drop.tip(smytree,smytree$tip.label[-match(smydata.j$species,
    smytree$tip.label)])

  smytree.j <- pruned.tree
  smydata.j <- smydata.j[match(smytree.j$tip.label,rownames(smydata.j)),]

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata.j, method = "REML")

  myout <- runpGls(smydata.j,smytree.j)
  jack.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(lambda.est, phy = smytree.j, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(l.est, phy = smytree.j, fixed = TRUE), method = "REML")
  }

  jStat.ols[,i] <- as.numeric(c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4]))
  jStat.pgls[,i] <- as.numeric(c(coef(model.pgls)[1],
    coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],
    coef(model.pgls)[3]+coef(model.pgls)[4]))
}
jackEst.ols <- t(t(apply(jStat.ols, 1, mean))) # jackknife estimate
jackEst.pgls <- t(t(apply(jStat.pgls, 1, mean))) # jackknife estimate

```

```

jack.lambdas # lambdas of the jackknifed models

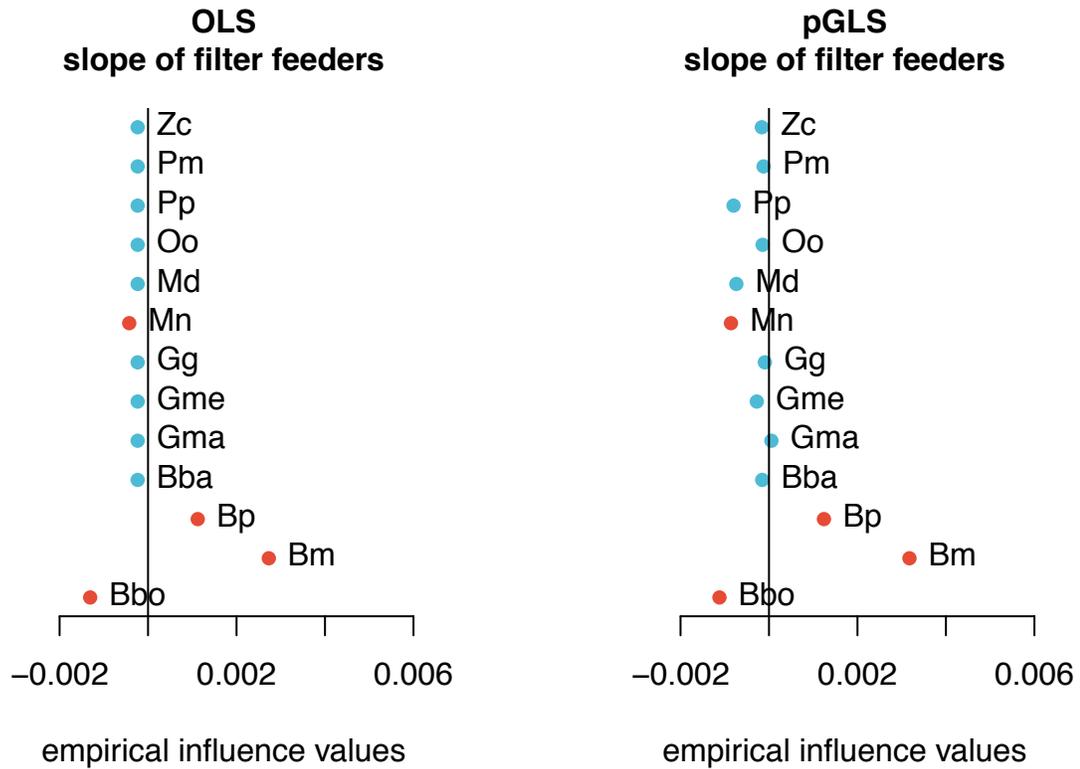
## [1] 0.8540419 0.8542140 0.8537932 0.8539142 0.8550103 0.8532686 0.8542298
## [8] 0.8538816 0.8508133 0.8539637 0.8504787 0.8540712 0.8538684

num.ols <- numeric(); den.ols <- numeric(); ahat.ols <- numeric()
num.pgls <- numeric(); den.pgls <- numeric(); ahat.pgls <- numeric()
for (i in 1:nrow(jStat.ols)){
  num.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^3 )
  den.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^2 )
  ahat.ols[i] <- num.ols[i]/(6*den.ols[i]^(3/2)) # ahat based on jackknife
  num.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^3 )
  den.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^2 )
  ahat.pgls[i] <- num.pgls[i]/(6*den.pgls[i]^(3/2)) # ahat based on jackknife
}

# influential species:
par(mfrow=c(1,2))
plot(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                    round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("OLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                    round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("pGLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```



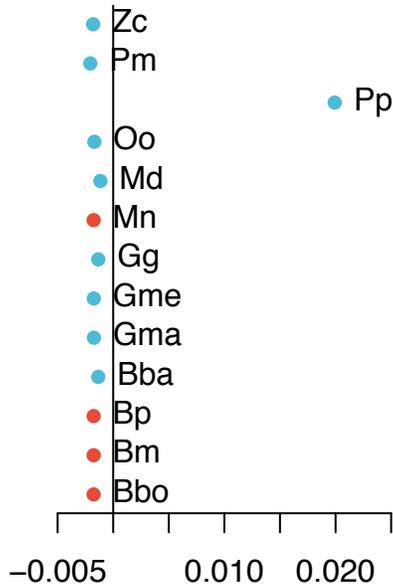
```

par(mfrow=c(1,2))
plot(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("OLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("pGLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

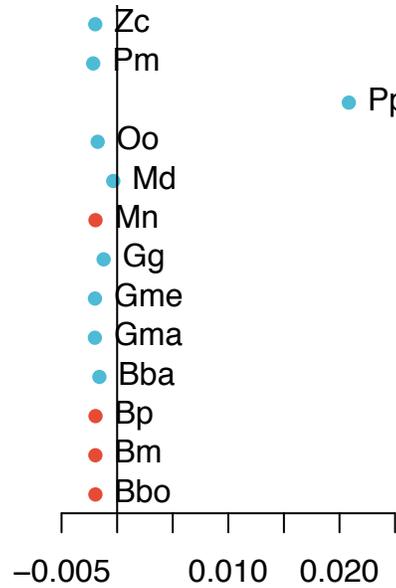
```

OLS
slope of single-prey feeders



empirical influence values

pGLS
slope of single-prey feeders



empirical influence values

```
# adjust quantiles for 100*(1-alpha)% bootstrap BCa interval
```

```
alpha <- 0.05
zL.ols <- z0.ols + qnorm(alpha/2)
alpha1.ols <- pnorm(z0.ols + zL.ols / (1-ahat.ols*zL.ols))
zU.ols <- z0.ols + qnorm(1-alpha/2)
alpha2.ols <- pnorm(z0.ols + zU.ols / (1-ahat.ols*zU.ols))

zL.pgls <- z0.pgls + qnorm(alpha/2)
alpha1.pgls <- pnorm(z0.pgls + zL.pgls / (1-ahat.pgls*zL.pgls))
zU.pgls <- z0.pgls + qnorm(1-alpha/2)
alpha2.pgls <- pnorm(z0.pgls + zU.pgls / (1-ahat.pgls*zU.pgls))
```

```
## new quantiles OLS:
cbind((alpha1.ols*100), (alpha2.ols*100))
```

```
##           [,1]      [,2]
## [1,]  0.578384341  93.99189
## [2,]  0.008529435  83.95515
## [3,]  6.461502218  99.51549
## [4,] 17.027959183  99.99384
```

```
## new quantiles pGLS:
cbind((alpha1.pgls*100), (alpha2.pgls*100))
```

```
##           [,1]      [,2]
## [1,]  0.621108317  94.23841
```

```

## [2,] 0.007569088 83.44803
## [3,] 6.128563310 99.46578
## [4,] 16.312332479 99.99197

CI.ols <- matrix(nrow = ncol(a.ols), ncol=2)
for (i in 1:ncol(a.ols)){
CI.ols[i,] <- quantile(a.ols[,i], c(alpha1.ols[i], alpha2.ols[i])) # BCa interval
}
df.boot.ols$lowerCIbca <- CI.ols[,1]
df.boot.ols$upperCIbca <- CI.ols[,2]

CI.pgls <- matrix(nrow = ncol(a.pgls), ncol=2)
for (i in 1:ncol(a.pgls)){
CI.pgls[i,] <- quantile(a.pgls[,i], c(alpha1.pgls[i], alpha2.pgls[i])) # BCa interval
}
df.boot.pgls$lowerCIbca <- CI.pgls[,1]
df.boot.pgls$upperCIbca <- CI.pgls[,2]

```

5.5.3 Plot OLS model

```

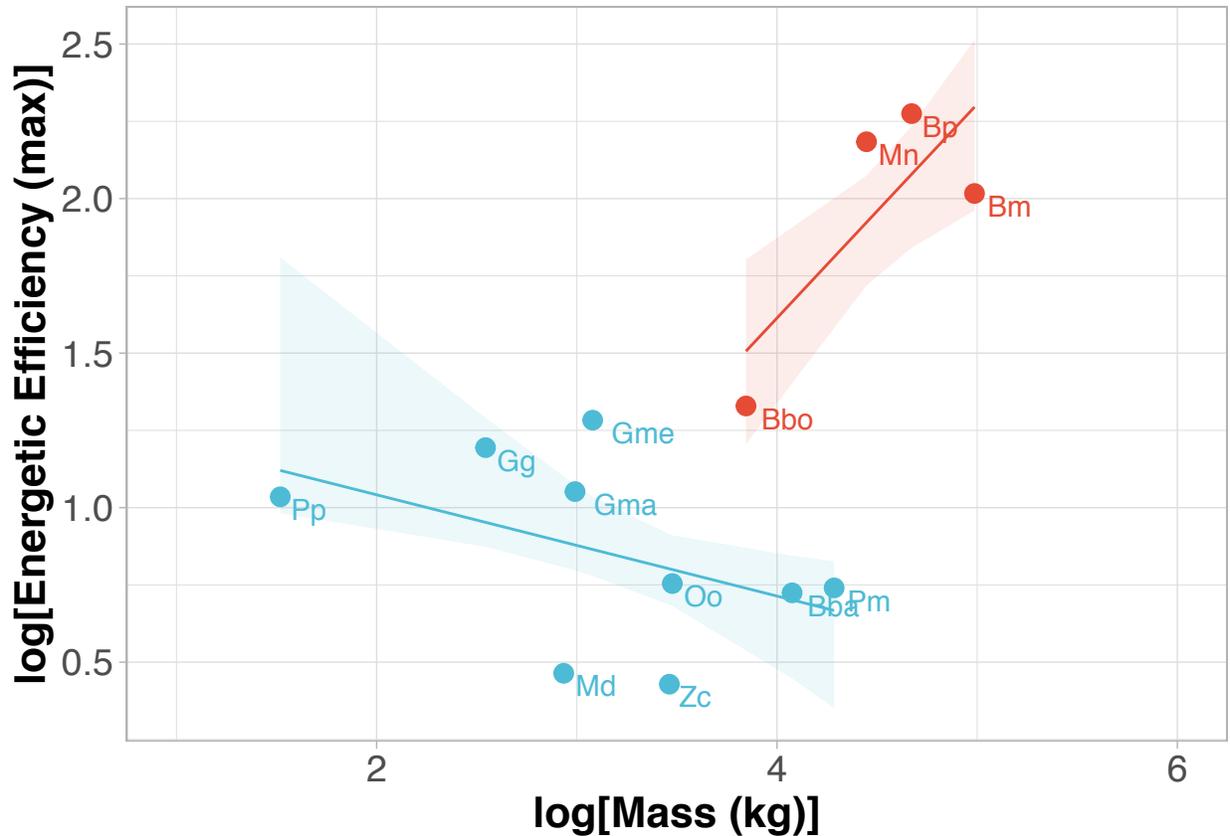
smydata <- smydata.orig

ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit,
                                     y_min = preds[1,], y_max = preds[2,]))

fig_ols <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
           linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
           linetype = 1) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Rorqual"), fill = "#E64B35FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Odontocete"), fill = "#4DBBD5FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
           aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
           aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_ols

```



5.5.3.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCI"],
  lowerCI.od = df.boot.ols["slope.od","lowerCI"],
  upperCI.od = df.boot.ols["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.ols["slope.od","lowerCIbca"],
  upperCI.od = df.boot.ols["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.ols[,3],
  odontocete_slope=a.ols[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF"), color = NA, alpha = 0.3) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF"), color = NA, alpha = 0.3) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

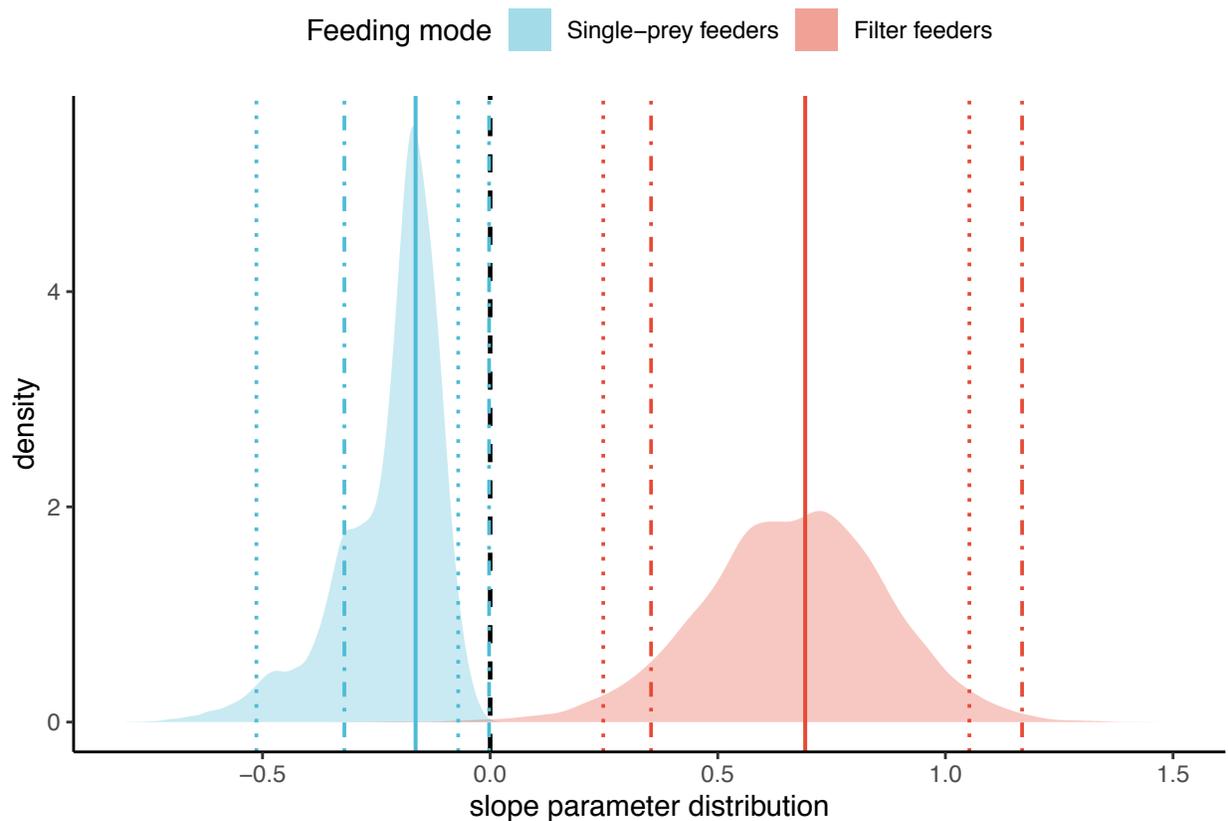
```

```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 1 rows containing non-finite values (stat_density).



```

rn <- rownames(df.boot.ols)
rownames(df.boot.ols) <- c("intercept filter", "intercept single-prey",

```

```

      "slope filter", "slope single-prey")
knitr::kable(df.boot.ols,
  caption = "OLS 95% Bootstrap Pctl and BCa CI",
  format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
  kable_styling(latex_options = "hold_position")

```

Table 3: OLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-1.1546	-1.0530	-2.7770	0.7939	-3.2920	0.3881
intercept single-prey	1.3691	1.5612	1.0986	2.5851	0.8356	1.9225
slope filter	0.6920	0.6654	0.2484	1.0526	0.3534	1.1684
slope single-prey	-0.1638	-0.2188	-0.5135	-0.0702	-0.3202	-0.0024

```
rownames(df.boot.ols) <- rn
```

5.5.4 Plot pGLS model

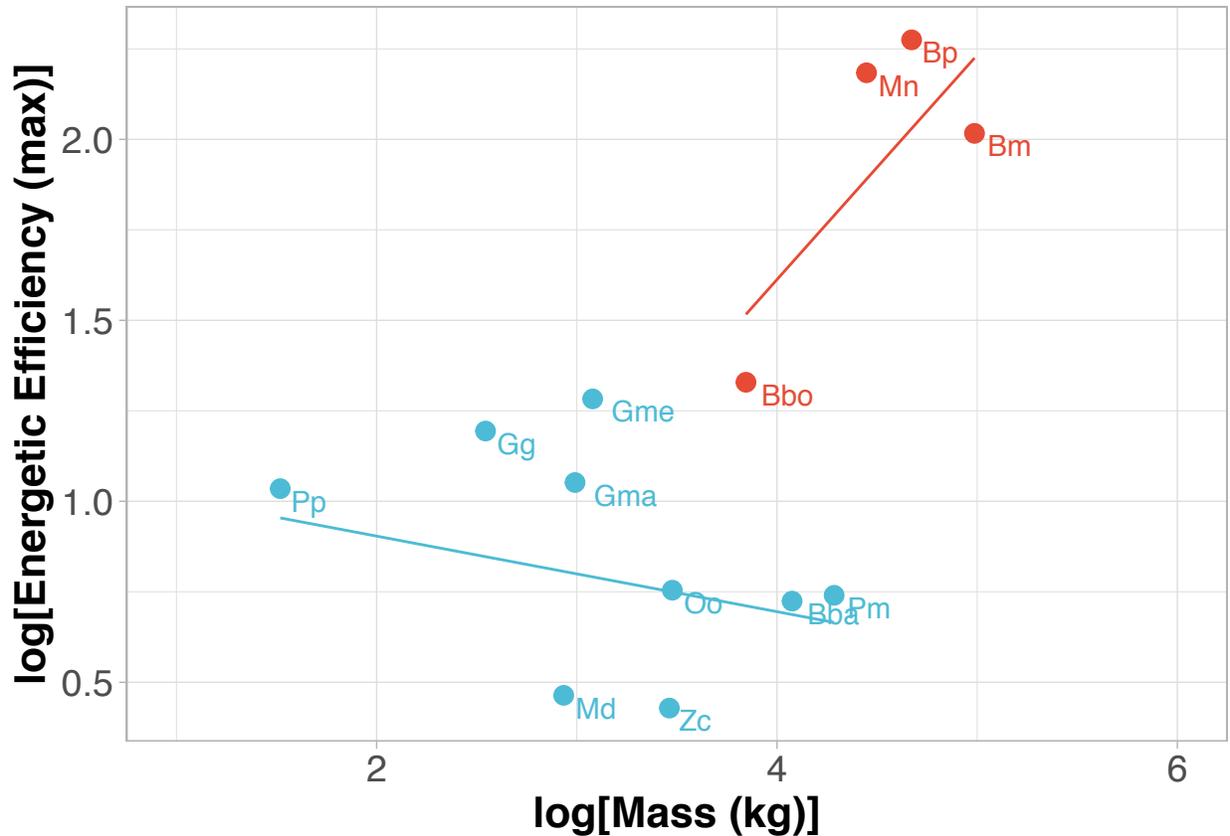
```

pgls.fit <- predict(m.pgls.nlme)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))

fig_pgls <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 1) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_pgls

```



5.5.4.1 Plot kernel density distributions of slopes

```

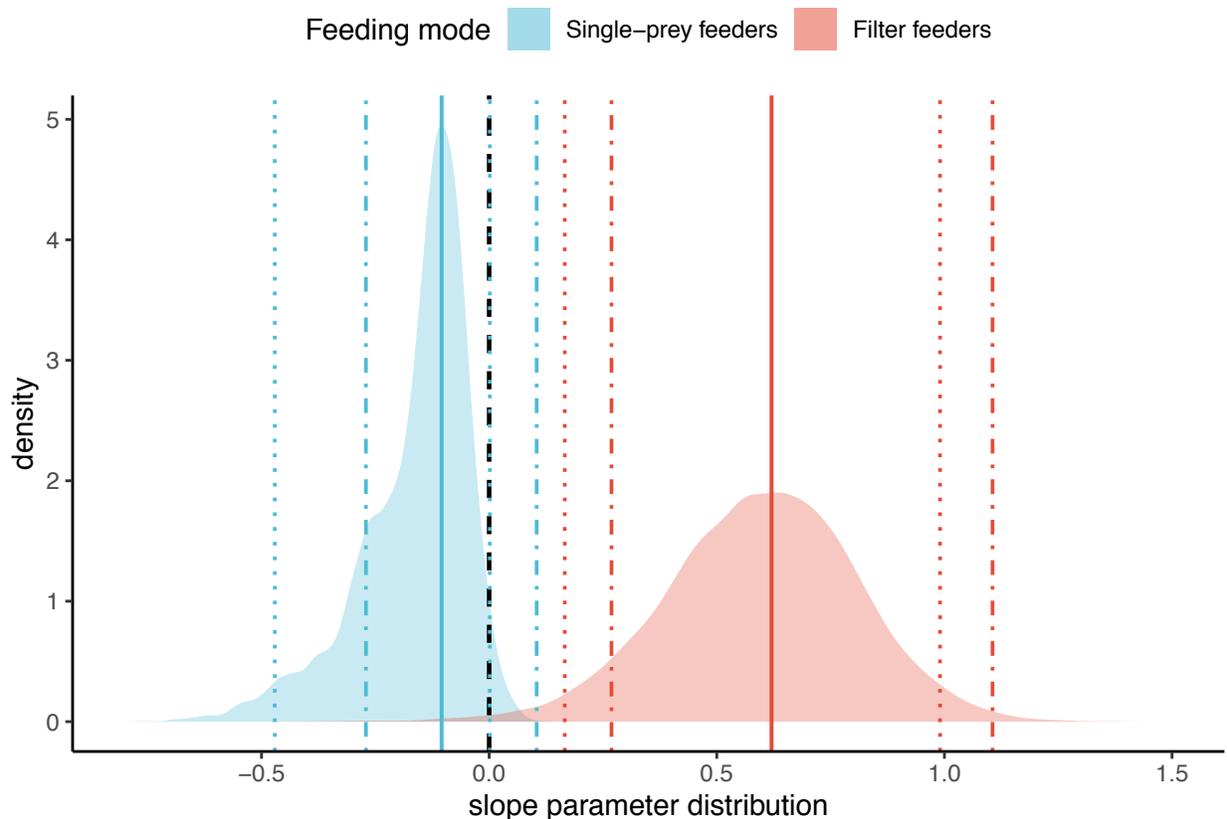
model_param <- data.frame(slope.rorq = df.boot.pgl["slope.rorq","obs"],
  slope.od = df.boot.pgl["slope.od","obs"],
  lowerCI.rorq = df.boot.pgl["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.pgl["slope.rorq","upperCI"],
  lowerCI.od = df.boot.pgl["slope.od","lowerCI"],
  upperCI.od = df.boot.pgl["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.pgl["slope.rorq","obs"],
  slope.od = df.boot.pgl["slope.od","obs"],
  lowerCI.rorq = df.boot.pgl["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.pgl["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.pgl["slope.od","lowerCIbca"],
  upperCI.od = df.boot.pgl["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.pgl[,3],
  odontocete_slope=a.pgl[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF", color = NA, alpha = 0.3)) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF", color = NA, alpha = 0.3)) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

```

```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```



```

rn <- rownames(df.boot.pgls)
rownames(df.boot.pgls) <- c("intercept filter","intercept single-prey",
  "slope filter","slope single-prey")
knitr::kable(df.boot.pgls,

```

```
caption = "pGLS 95% Bootstrap Pctl and BCa CI",
format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
kable_styling(latex_options = "hold_position")
```

Table 4: pGLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-0.8687	-0.7809	-2.5529	1.1440	-3.0660	0.7069
intercept single-prey	1.1126	1.3293	0.7812	2.4723	0.4201	1.7249
slope filter	0.6203	0.5973	0.1661	0.9904	0.2687	1.1057
slope single-prey	-0.1044	-0.1613	-0.4710	0.0014	-0.2707	0.1043

```
rownames(df.boot.pgls) <- rn
```

5.6 Extract summary statistics

```
specify_decimal <- function(x, k) trimws(format(round(x, k), nsmall = k))

res.df.ols <- m.ols$dimN - m.ols$dimP

res.df.pgls <- m.pgls.nlme$dimN - m.pgls.nlme$dimP

intercepts.od.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCIbca"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCIbca"], 4),
  ")"))

intercepts.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCIbca"],
  4), ")"),
  paste0(specify_decimal(df.boot.ols["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.rorq", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.rorq", "lowerCIbca"], 4),
```

```

" - ", specify_decimal(df.boot.ols["intercept.rorq", "upperCIbca"], 4),
")"))

slopes.od.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.od", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.pgl["slope.od", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.pgl["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.od", "lowerCIbca"], 4), " - ",
  specify_decimal(df.boot.pgl["slope.od", "upperCIbca"], 4), ")"),
paste0(specify_decimal(df.boot.ols["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.od", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.ols["slope.od", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.ols["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.od", "lowerCIbca"], 4), " - ",
  specify_decimal(df.boot.ols["slope.od", "upperCIbca"], 4), ")"))

slopes.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.rorq", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.pgl["slope.rorq", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.pgl["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgl["slope.rorq", "upperCIbca"],
  4), ")"),
paste0(specify_decimal(df.boot.ols["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.rorq", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.ols["slope.rorq", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.ols["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.ols["slope.rorq", "upperCIbca"],
  4), ")"))

a.od.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.pgl["intercept.od", "lowerCI"]), 4),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.od", "upperCI"]), 4),
  ")"),
paste0(specify_decimal(10^(df.boot.pgl["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.pgl["intercept.od", "lowerCIbca"]), 4),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.od", "upperCIbca"]), 4),
  ")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.ols["intercept.od", "lowerCI"]), 4), " - ",
  specify_decimal(10^(df.boot.ols["intercept.od", "upperCI"]), 4), ")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.ols["intercept.od", "lowerCIbca"]), 4),
  " - ", specify_decimal(10^(df.boot.ols["intercept.od", "upperCIbca"]),
  4), ")"))

a.rorq.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq", "obs"]), 4),
  " (", specify_decimal(10^(df.boot.pgl["intercept.rorq", "lowerCI"]), 5),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.rorq", "upperCI"]), 4), ")"),
paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq", "obs"]), 4),
  " (", specify_decimal(10^(df.boot.pgl["intercept.rorq", "lowerCIbca"]), 5),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.rorq", "upperCIbca"]), 4),
  ")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.rorq", "obs"]), 4),

```

Table 5: Model summary statistics

	<i>Filter feeders</i>		<i>Single-prey feeders</i>		RSE	tot.df	res.df
	slope*	intercept	slope	intercept			
pGLS	0.6203 (0.1661 - 0.9904)	-0.8687 (-2.5529 - 1.1440)	-0.1044 (-0.4710 - 0.0014)	1.1126 (0.7812 - 2.4723)	0.3616		
	0.6203 (0.2687 - 1.1057)	-0.8687 (-3.0660 - 0.7069)	-0.1044 (-0.2707 - 0.1043)	1.1126 (0.4201 - 1.7249)	0.3616		
OLS	0.6920 (0.2484 - 1.0526)	-1.1546 (-2.7770 - 0.7939)	-0.1638 (-0.5135 - -0.0702)	1.3691 (1.0986 - 2.5851)	0.3008	13	9
	0.6920 (0.3534 - 1.1684)	-1.1546 (-3.2920 - 0.3881)	-0.1638 (-0.3202 - -0.0024)	1.3691 (0.8356 - 1.9225)	0.3008		

Note:

* Throughout the table, values in brackets represent 95% confidence intervals: percentile in shaded rows, BCa in non-shaded rows.

```

" (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCI"]),5),
" - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCI"]),4),")",
paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCIbca"]),5),
" - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCIbca"]),4),
")"))

RSE <- rbind(specify_decimal(t(t(rep(as.numeric(m.pgls.nlm$sigma),2))),4),
             specify_decimal(t(t(rep(as.numeric(m.ols$sigma),2))),4))
df <- cbind(t(t(c(rep(m.pgls.nlm$dims$N,2), rep(m.ols$dims$N,2)))),
            t(t(c(rep(res.df.pgls,2), rep(res.df.ols,2)))))
models <- rbind(t(t(rep("pGLS",2))),t(t(rep("OLS",2))))

outputs <- cbind(models, slopes.rorq.ci, intercepts.rorq.ci, slopes.od.ci,
                 intercepts.od.ci, RSE, df)
df.outputs <- data.frame(outputs, check.rows = TRUE, check.names = TRUE)
names(df.outputs) <- c("", "slope", "intercept", "slope", "intercept", "RSE", "tot.df", "res.df")
names(df.outputs)[2] <- paste0(names(df.outputs)[2],
                              footnote_marker_symbol(1))

knitr::kable(df.outputs,
             caption = "Model summary statistics",
             format = "latex", booktabs = TRUE, escape = FALSE) %>%
kable_styling(latex_options = "scale_down") %>%
row_spec(0, bold = T) %>%
row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
column_spec(c(1,(ncol(df.outputs)-1):ncol(df.outputs))-1,
            background = "white") %>%
column_spec(1, bold = T) %>%
collapse_rows(columns = c(1,(ncol(df.outputs)-1):ncol(df.outputs))) %>%
add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2,
                  " " = 3), bold = T, italic = T) %>%
footnote(general = "", general_title = "Note:",
         symbol = paste0("Throughout the table, values in brackets",
                        " represent 95% confidence intervals: ",
                        "percentile in shaded rows, BCa in non-shaded rows."),
         symbol_title = "", title_format = "italic",
         footnote_as_chunk = T)

alloout <- cbind(models, a.rorq.ci, slopes.rorq.ci, a.od.ci, slopes.od.ci)
df.allo <- data.frame(alloout, check.rows = TRUE, check.names = TRUE)
names(df.allo) <- c("", "a", "b", "a", "b")
names(df.allo)[2] <- paste0(names(df.allo)[2], footnote_marker_symbol(1))
knitr::kable(df.allo,

```

Table 6: Transformed to allometric equations

	<i>Filter feeders</i>		<i>Single-prey feeders</i>	
	a*	b	a	b
pGLS	0.1353 (0.00280 - 13.9321)	0.6203 (0.1661 - 0.9904)	12.9595 (6.0426 - 296.6547)	-0.1044 (-0.4710 - 0.0014)
	0.1353 (0.00086 - 5.0916)	0.6203 (0.2687 - 1.1057)	12.9595 (2.6309 - 53.0772)	-0.1044 (-0.2707 - 0.1043)
OLS	0.0700 (0.00167 - 6.2221)	0.6920 (0.2484 - 1.0526)	23.3942 (12.5489 - 384.6443)	-0.1638 (-0.5135 - -0.0702)
	0.0700 (0.00051 - 2.4438)	0.6920 (0.3534 - 1.1684)	23.3942 (6.8488 - 83.6489)	-0.1638 (-0.3202 - -0.0024)

* Throughout the table, values in brackets represent 95% confidence intervals.: percentile in shaded rows, BCa in non-shaded rows.

```
caption = "Transformed to allometric equations",
format = "latex", booktabs = TRUE, escape = FALSE) %>%
kable_styling(latex_options = "scale_down") %>%
row_spec(0, bold = T) %>%
row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
column_spec(1, bold = T) %>%
collapse_rows(columns = 1) %>%
add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2),
bold = T, italic = T) %>%
footnote(symbol = paste0("Throughout the table, values in brackets",
" represent 95% confidence intervals.: ",
"percentile in shaded rows, BCa in non-shaded rows."),
symbol_title = "", threeparttable = TRUE, footnote_as_chunk = T)
```

5.7 Plot best models (OLS - dashed, PGLS - solid)

```
pgls.fit <- predict(m.pgls.nlme)
ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))

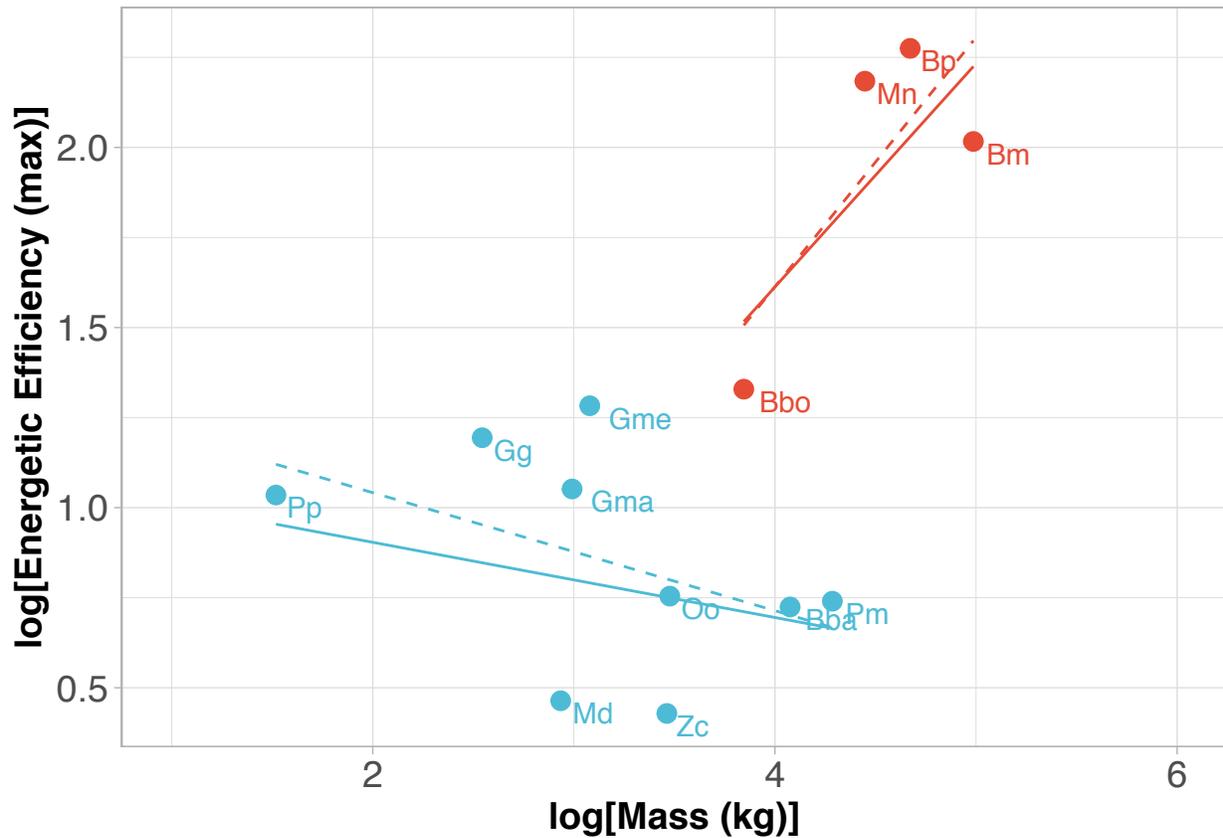
fig_4.45 <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe2, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 2) +
  geom_line(data = dplyr::filter(predframe2, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 2) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  xlim(1,6) +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 14,
    face = "bold")) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
```

```

geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
          aes(label = abbreviation, hjust = -0.3, vjust = 1.1) +
geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
          aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

```

fig_4.45



5.7.1 Construct output table

```

df.out <- smydata[,c("species", "fm", "x_mean", "y_mean")]
df.out$fitted_ols <- fitted(m.ols)
df.out$fitted_pgls <- fitted(m.pgls.nlme)
rownames(df.out) <- NULL
kable(df.out,
      caption = "Model outputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")

```

5.8 Quick clean up

```

m.45.pgls.nlme <- m.pgls.nlme
df.45.outputs <- df.outputs
m.45.ols <- m.ols
to.keep <- c("df.spec", "mytree", "d_full", "m.45.pgls.nlme", "df.45.outputs", "m.45.ols",
            "abbreviation")

```

Table 7: Model outputs

species	fm	x_mean	y_mean	fitted_ols	fitted_pgl
Balaenoptera_bonaerensis	Filter	3.8451	1.3289	1.5062	1.5165
Balaenoptera_musculus	Filter	4.9868	2.0165	2.2963	2.2247
Balaenoptera_physalus	Filter	4.6725	2.2749	2.0788	2.0298
Berardius_bairdii	Single-prey	4.0755	0.7242	0.7015	0.6871
Globicephala_macrorhynchus	Single-prey	2.9912	1.0519	0.8791	0.8003
Globicephala_melas	Single-prey	3.0792	1.2829	0.8647	0.7912
Grampus_griseus	Single-prey	2.5441	1.1941	0.9524	0.8470
Megaptera_novaeangliae	Filter	4.4472	2.1839	1.9229	1.8900
Mesoplodon_densirostris	Single-prey	2.9345	0.4639	0.8884	0.8063
Orcinus_orca	Single-prey	3.4771	0.7544	0.7996	0.7496
Phocoena_phocoena	Single-prey	1.5185	1.0350	1.1204	0.9541
Physeter_macrocephalus	Single-prey	4.2856	0.7403	0.6671	0.6652
Ziphius_cavirostris	Single-prey	3.4624	0.4286	0.8020	0.7511

```
to.keep <- c(to.keep, "to.keep")
rm(list=setdiff(ls(), to.keep))
```

6 Run model for MR = .61

6.1 Prepare data

6.1.1 Get rid of rows with NAs - subset the data

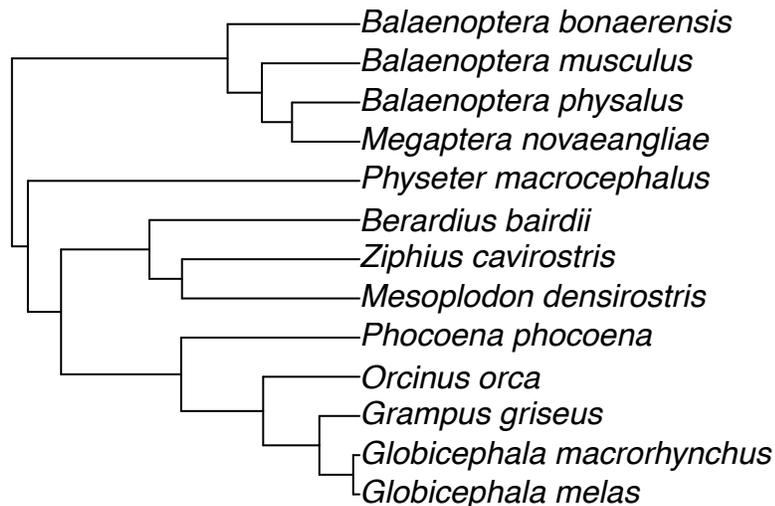
```
smydata <- df.spec
smydata$y_mean <- smydata$wgtMean.61
smydata <- smydata[!is.na(smydata$y_mean),]
smydata <- smydata[!is.na(smydata$x_mean),]
smydata$fm <- factor(smydata$fm)
smydata$Group <- smydata$Group
colnames(smydata)[1] <- "species"
```

6.1.2 Adjust tree - drop species for which data are missing

```
smytree <- drop.tip(mytree, mytree$tip.label[-match(smydata$species, mytree$tip.label)])
plot(smytree)
```

Table 8: Model inputs

species	gr	x_mean	fm	Group	abbreviation	wgtMean.45	wgtMean.61	wgtMean.68	wgtMean.75	y_mean
<i>Balaenoptera bonaerensis</i>	1	3.8451	Filter	Rorqual	Bbo	1.3289	1.2293	1.1386	1.0061	1.2293
<i>Balaenoptera musculus</i>	1	4.9868	Filter	Rorqual	Bm	2.0165	1.9307	1.8293	1.6606	1.9307
<i>Balaenoptera physalus</i>	1	4.6725	Filter	Rorqual	Bp	2.2749	2.1286	2.0246	1.8348	2.1286
<i>Berardius bairdii</i>	5	4.0755	Single-prey	Odontocete	Bba	0.7242	-0.3180	-0.0448	-0.3180	-0.3180
<i>Globicephala macrorhynchus</i>	2	2.9912	Single-prey	Odontocete	Gma	1.0519	0.7918	0.6401	0.4703	0.7918
<i>Globicephala melas</i>	2	3.0792	Single-prey	Odontocete	Gme	1.2829	0.8406	0.6335	0.4257	0.8406
<i>Grampus griseus</i>	2	2.5441	Single-prey	Odontocete	Gg	1.1941	0.8068	0.6328	0.4577	0.8068
<i>Megaptera novaeangliae</i>	1	4.4472	Filter	Rorqual	Mn	2.1839	2.0144	1.8621	1.6563	2.0144
<i>Mesoplodon densirostris</i>	5	2.9345	Single-prey	Odontocete	Md	0.4639	0.0020	-0.2024	-0.4092	0.0020
<i>Orcinus orca</i>	2	3.4771	Single-prey	Odontocete	Oo	0.7544	0.6359	0.5393	0.4096	0.6359
<i>Phocoena phocoena</i>	3	1.5185	Single-prey	Odontocete	Pp	1.0350	0.8313	0.7372	0.6407	0.8313
<i>Physeter macrocephalus</i>	4	4.2856	Single-prey	Odontocete	Pm	0.7403	0.3781	0.1685	-0.0732	0.3781
<i>Ziphius cavirostris</i>	5	3.4624	Single-prey	Odontocete	Zc	0.4286	-0.0079	-0.2229	-0.4496	-0.0079



6.1.3 Rearrange the row order in smydata to match smytree:

```
smydata <- smydata[match(smytree$tip.label, rownames(smydata)),]
rownames(smydata) <- NULL
kable(smydata,
      caption = "Model inputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")
```

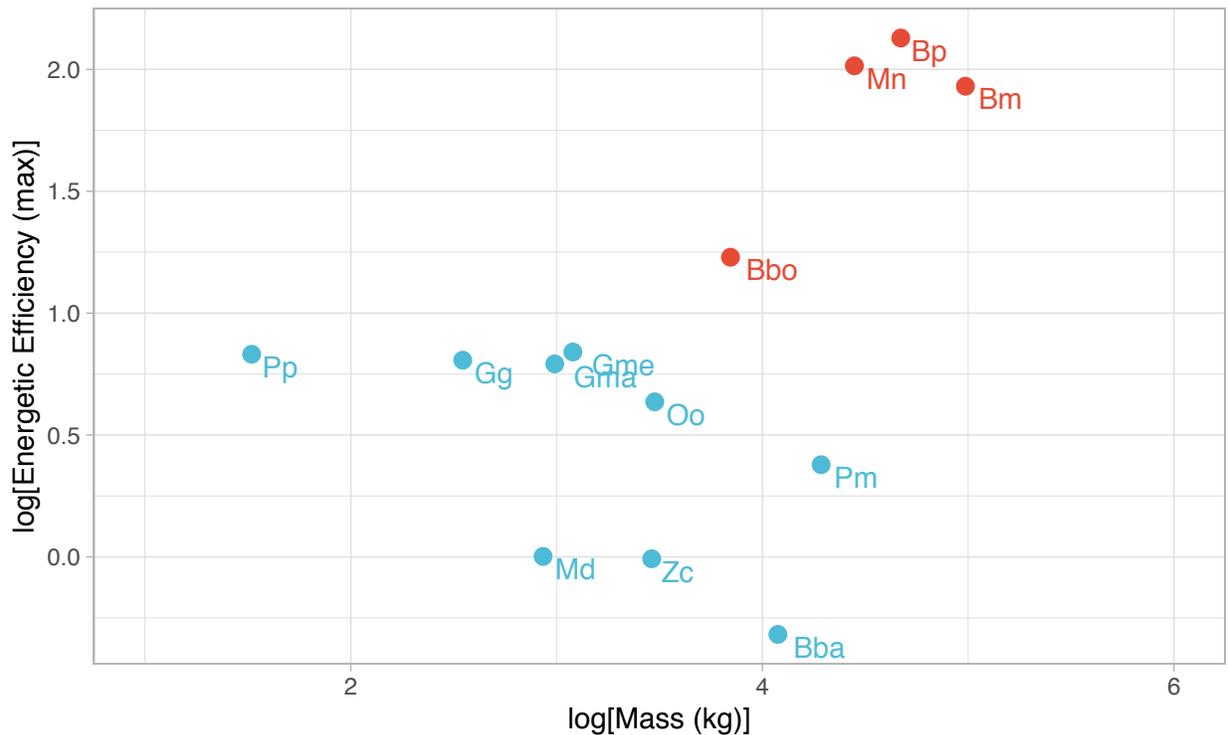
```
rownames(smydata) <- smydata$species
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_61_smydata.rds")
```

```
saveRDS(smydata,fil)
```

6.2 Plot the data

```
ggplot(smydata, aes(x_mean, y = value, color = Group)) +  
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), shape = 16, size = 3,  
            aes(y = y_mean, color = "#4DBBD5FF")) +  
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), shape = 16, size = 3,  
            aes(y = y_mean, color = "#E64B35FF")) +  
  scale_color_manual(name = "",  
                    values = c("#E64B35FF", "#4DBBD5FF"),  
                    labels = c("Filter feeders", "Single-prey feeders")) +  
  theme_light() + theme(legend.position = "top") +  
  xlim(1,6) +  
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +  
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",  
           aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1) +  
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",  
           aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1)
```

● Filter feeders ● Single-prey feeders



6.3 Run OLS with feeding mode as a categorical predictor

6.3.1 Run OLS and model reduction using ML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "ML")
summary(m.ols)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 16.03576 18.8605 -3.017879
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.2868747 1.9791410 -0.6502188 0.5318
## fmSingle-prey  2.7078826 2.0437555  1.3249543 0.2178
## x_mean        0.6935622 0.4390983  1.5795144 0.1487
## fmSingle-prey:x_mean -1.0047736 0.4663325 -2.1546290 0.0596
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.6572547 -0.7897083  0.5725294  0.9530080  1.2379850
##
## Residual standard error: 0.3051982
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.ols)
```

```
## Denom. DF: 9
##      numDF F-value p-value
## (Intercept)  1 72.53500 <.0001
## fm          1 39.52090 0.0001
## x_mean      1  1.78017 0.2149
## fm:x_mean   1  4.64243 0.0596
```

```
m.ols.2 <- update(m.ols, ~ . - fm:x_mean)
anova(m.ols, m.ols.2)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## m.ols      1  5 16.03576 18.86050 -3.017879
## m.ols.2    2  4 19.44324 21.70303 -5.721618 1 vs 2 5.407479 0.0201
```

6.3.1.1 Compare to an intercept-only model

```
m.ols.0 <- gls(y_mean ~ 1, data = smydata, method = "ML")
anova(m.ols, m.ols.0)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## m.ols      1  5 16.03576 18.86050 -3.017879
## m.ols.0    2  2 33.55381 34.68370 -14.776903 1 vs 2 23.51805 <.0001
```

```
m.ols.p <- anova(m.ols, m.ols.0)$`p-value`[2]
```

6.3.2 Estimate final model using REML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
summary(m.ols)
```

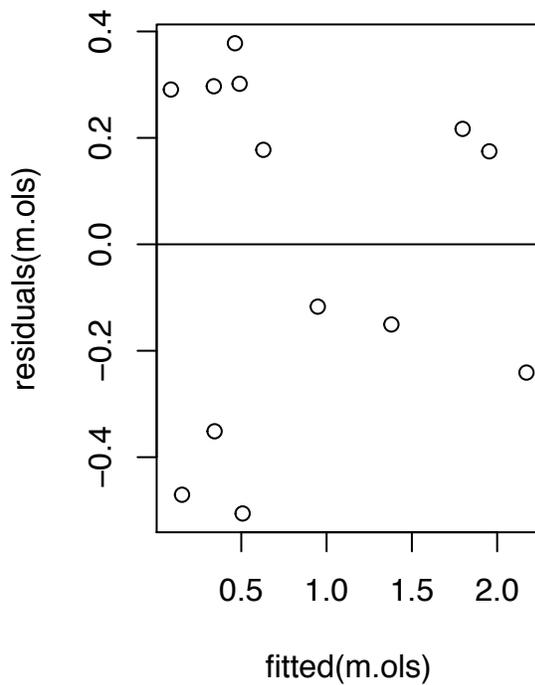
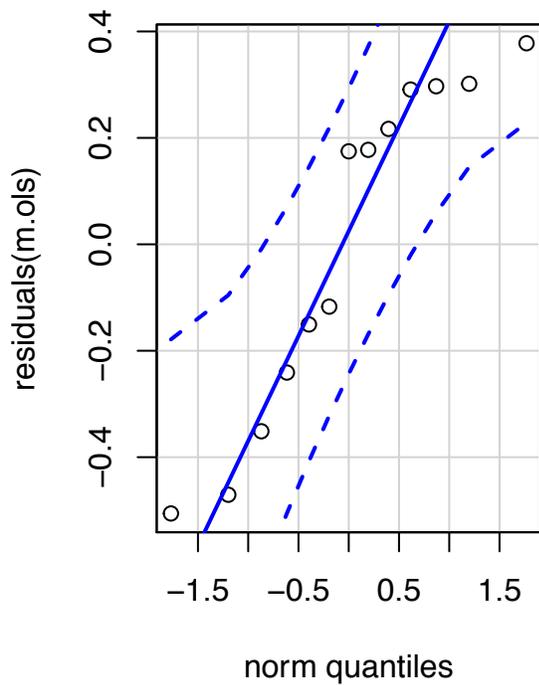
```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 22.40861 23.39473 -6.204303
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.2868747  1.9791410  -0.6502188  0.5318
## fmSingle-prey  2.7078826  2.0437555   1.3249543  0.2178
## x_mean        0.6935622  0.4390983   1.5795144  0.1487
## fmSingle-prey:x_mean -1.0047736  0.4663325  -2.1546290  0.0596
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.3789193 -0.6570770  0.4763732  0.7929506  1.0300658
##
## Residual standard error: 0.3668025
## Degrees of freedom: 13 total; 9 residual
```

```
m.ols.param <- as.data.frame(t(summary(m.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.ols.param <- m.ols.param[5:8]
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_61_m_ols_param.rds")
saveRDS(m.ols.param,fil)
```

6.3.2.1 Model diagnostics

6.3.2.1.1 QQ-plot and Residuals vs fitted plot

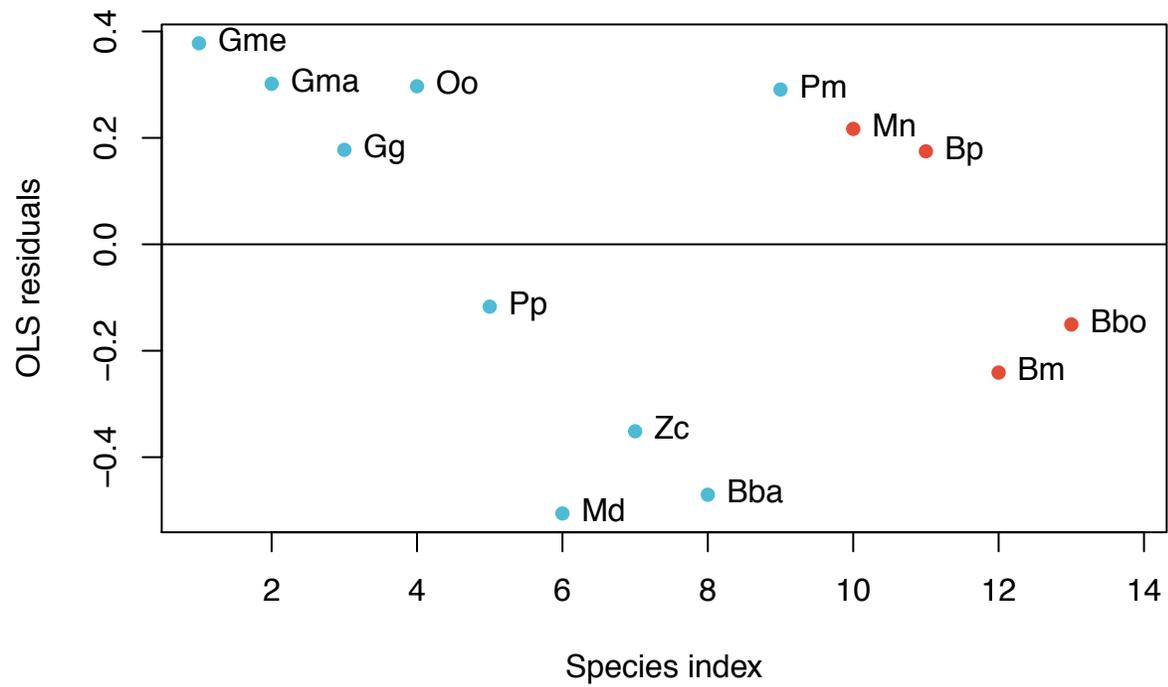
```
par(mfrow=c(1,2))
qqPlot(residuals(m.ols), id=FALSE)
plot(fitted(m.ols), residuals(m.ols))
abline(0,0)
```



6.3.3 Evaluate for phylogenetic correlation

6.3.3.1 Plot residuals ordered “by phylogeny” (i.e. in the order of tips of the phylogenetic tree)

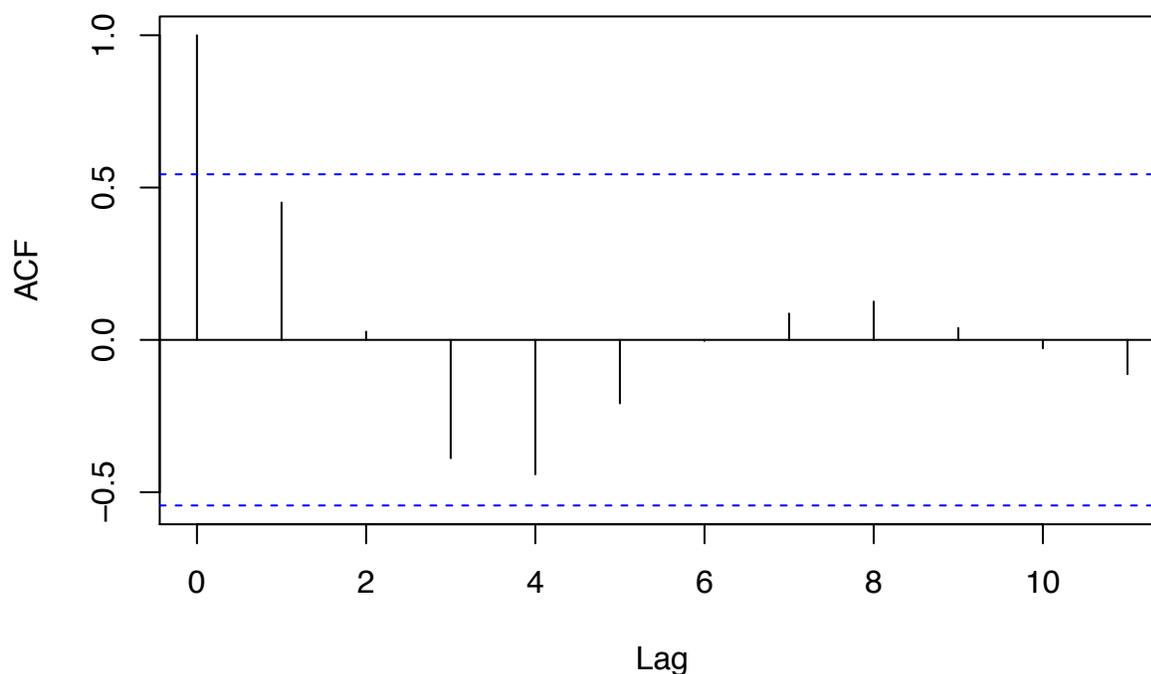
```
is_tip <- smytree$edge[,2]<=length(smytree$tip.label)
ordered_tips <- smytree$edge[is_tip,2] # extract the order of tree tips
oj <- residuals(m.ols)
tl <- smytree$tip.label[ordered_tips]
res <- oj[tl]
plot(oj[tl], pch=16, ylab="OLS residuals", xlab="Species index",
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata[tl,"fm"])],
     xlim=c(1,13.8))
abline(0,0)
text(oj[tl], labels=abbreviation[tl], pos=4)
```



6.3.3.2 Plot autocorrelation function of residuals ordered “by phylogeny”

```
acf(res, main="Series: residuals sorted by phylogeny")
```

Series: residuals sorted by phylogeny



6.4 Run a pGLS with feeding mode as a categorical predictor

6.4.1 Estimate Pagel's λ (amount of phylogenetic signal) for each trait separately

```
lambdax <- phylosig(smytree, smydata$x_mean, method = "lambda", test = T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
lambday <- phylosig(smytree, smydata$y_mean, method = "lambda", test = T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
cbind(lambdax, lambday)

##      lambdax      lambday
## lambda 1.014327  1.015057
## logL   -12.69024 -5.793493
## logL0  -17.41681 -14.7769
## P       0.002107892 2.247893e-05
```

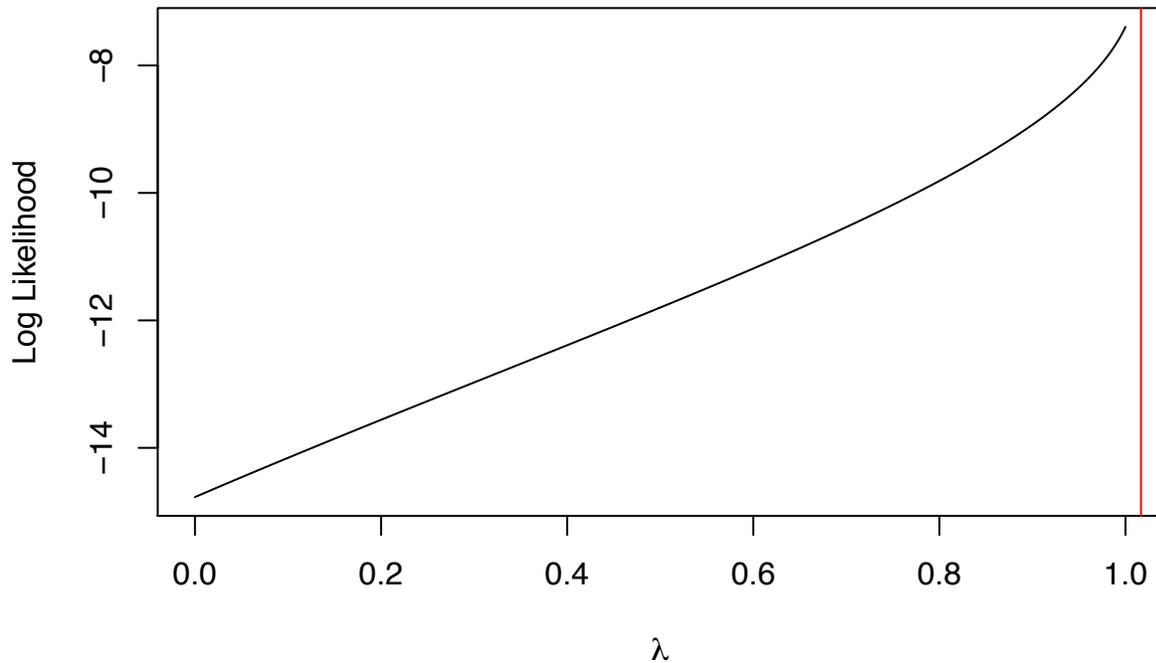
6.4.2 Plot likelihood surface for Pagel's λ for model without feeding mode as a covariate

λ estimate for the model marked in red.

```
lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ x_mean, smydata,
  method = "REML", correlation = corPagel(value = lambda, phy = smytree,
  fixed = TRUE))))
```

```
plot(lik ~ lambda, type = "l", main =
      expression(paste("Prey energy to body mass Likelihood Plot for ", lambda)),
      ylab = "Log Likelihood", xlab = expression(lambda))
m.pa.only <- gls(y_mean ~ x_mean, data = smydata, correlation =
                 corPagel(value = 0, phy = smytree, fixed = FALSE), method = "REML")
abline(v = m.pa.only$modelStruct[1], col = "red")
```

Prey energy to body mass Likelihood Plot for λ



6.4.3 Estimate Pagel's λ using REML

If λ is estimated to be greater than 1, fix it at 1, if smaller than 0, fix it at 0.

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
                  corPagel(1, phy = smytree, fixed = FALSE), method = "REML")
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 14.7419 15.92525 -1.370951
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
## 1.005418
##
```

```
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -0.9245974 1.2648694 -0.7309825 0.4834
## fmSingle-prey  1.7741186 1.3529485  1.3112980 0.2222
## x_mean        0.6025697 0.2767259  2.1774964 0.0574
## fmSingle-prey:x_mean -0.7611559 0.3051209 -2.4946047 0.0342
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.935
## x_mean         -0.966  0.903
## fmSingle-prey:x_mean 0.876 -0.953 -0.907
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.3785363 -0.4311878 0.5887827 0.8935836 1.2679680
##
## Residual standard error: 0.3780514
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.pgls.nlme)
```

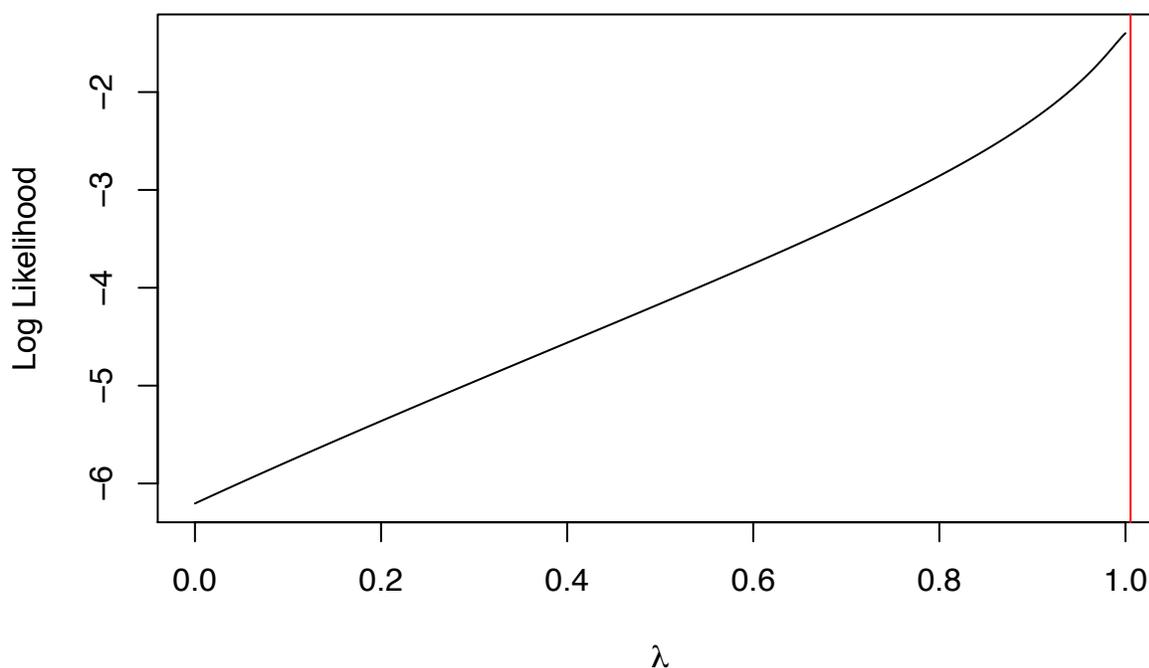
```
## Denom. DF: 9
##          numDF  F-value p-value
## (Intercept)    1 16.800257 0.0027
## fm             1 13.266273 0.0054
## x_mean         1  0.040672 0.8447
## fm:x_mean      1  6.223053 0.0342
```

```
lambda.est <- as.numeric(m.pgls.nlme$modelStruct[1])
if(lambda.est > 1){lambda.est <- 1} else if(lambda.est < 0){lambda.est <- 0}
```

6.4.4 Plot likelihood surface for Pagel's λ - our estimate marked in red

```
lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ fm * x_mean, smydata,
                                                method = "REML", correlation =
                                                corPagel(value = lambda, phy = smytree, fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
     expression(paste("Energetic Efficiency to Body mass Likelihood Plot for ", lambda)),
     ylab = "Log Likelihood", xlab = expression(lambda)),
abline(v = m.pgls.nlme$modelStruct, col = "red")
```

Energetic Efficiency to Body mass Likelihood Plot for λ



6.4.5 Run pGLS and model reduction with a fixed Pagel's λ (using ML)

```
m.pgls.nlmf <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
                  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")  
summary(m.pgls.nlmf)
```

```
## Generalized least squares fit by maximum likelihood  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
##      AIC      BIC  logLik  
## 7.201335 10.02608 1.399333  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
##      1  
##  
## Coefficients:  
##              Value Std.Error   t-value p-value  
## (Intercept)   -0.9301408 1.2460336 -0.7464814 0.4744  
## fmSingle-prey  1.8072006 1.3331935  1.3555427 0.2083  
## x_mean         0.6039625 0.2726807  2.2149077 0.0540  
## fmSingle-prey:x_mean -0.7706887 0.3008796 -2.5614521 0.0306  
##  
## Correlation:
```

```

##          (Intr) fmSng- x_mean
## fmSingle-prey      -0.935
## x_mean             -0.966  0.903
## fmSingle-prey:x_mean  0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.6714597 -0.5279175  0.6990477  1.0977839  1.5461811
##
## Residual standard error: 0.3084261
## Degrees of freedom: 13 total; 9 residual
anova(m.pgls.nlme)

## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)    1 17.560023  0.0023
## fm              1 13.861211  0.0047
## x_mean         1  0.063462  0.8068
## fm:x_mean      1  6.561037  0.0306

m.pgls.nlme.2 <- update(m.pgls.nlme, ~ . - fm:x_mean)
anova(m.pgls.nlme, m.pgls.nlme.2)

##      Model df      AIC      BIC   logLik  Test  L.Ratio
## m.pgls.nlme      1  5  7.201335 10.02608  1.399333
## m.pgls.nlme.2    2  4 12.319427 14.57922 -2.159714 1 vs 2  7.118092
##      p-value
## m.pgls.nlme
## m.pgls.nlme.2  0.0076

m.pgls.fm <- gls(y_mean ~ fm, data = smydata, correlation =
                corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")
summary(m.pgls.fm)

## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm
## Data: smydata
##      AIC      BIC   logLik
## 10.37234 12.06719 -2.186169
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##      Value Std.Error  t-value p-value
## (Intercept)  1.735839 0.3840132  4.520260  0.0009
## fmSingle-prey -1.419874 0.4545232 -3.123877  0.0097
##
## Correlation:
##      (Intr)
## fmSingle-prey -0.845
##

```

```
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.5599342 -0.7726679  0.6854618  1.1707976  1.2909025
##
## Residual standard error: 0.4063806
## Degrees of freedom: 13 total; 11 residual
```

```
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 7.201335 10.02608 1.399333
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -0.9301408 1.2460336 -0.7464814  0.4744
## fmSingle-prey  1.8072006 1.3331935  1.3555427  0.2083
## x_mean         0.6039625 0.2726807  2.2149077  0.0540
## fmSingle-prey:x_mean -0.7706887 0.3008796 -2.5614521  0.0306
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey   -0.935
## x_mean          -0.966  0.903
## fmSingle-prey:x_mean  0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.6714597 -0.5279175  0.6990477  1.0977839  1.5461811
##
## Residual standard error: 0.3084261
## Degrees of freedom: 13 total; 9 residual
```

6.4.5.1 Compare to an intercept-only model

```
m.pgls.nlme.0 <- gls(y_mean ~ 1, smydata, correlation = corPagel(value = lambda.est,
  phy = smytree, fixed = TRUE), method = "ML")
anova(m.pgls.nlme, m.pgls.nlme.0)
```

```
##      Model df      AIC      BIC    logLik   Test  L.Ratio
## m.pgls.nlme      1  5  7.201335 10.02608  1.399333
## m.pgls.nlme.0    2  2 16.628190 17.75809 -6.314095 1 vs 2 15.42685
##
##      p-value
## m.pgls.nlme
## m.pgls.nlme.0  0.0015
```

```
m.pgls.p <- anova(m.pgls.nlme, m.pgls.nlme.0)$`p-value`[2]
```

6.4.6 Estimate final model using REML

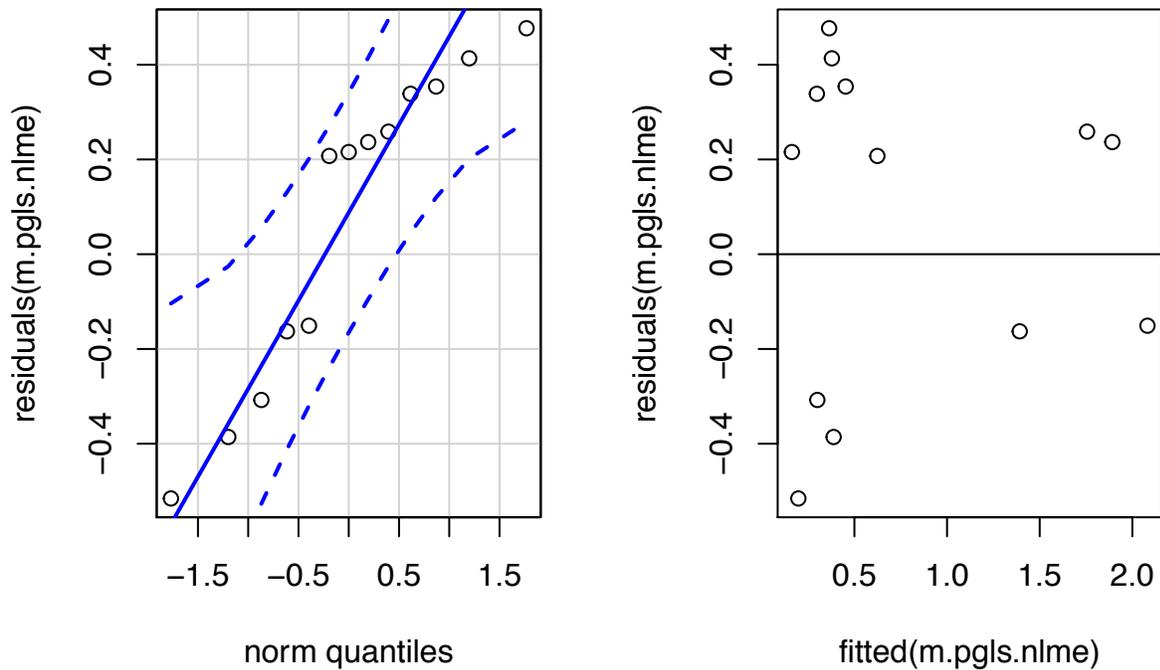
```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
                 corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "REML")  
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by REML  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
##      AIC      BIC    logLik  
## 12.79766 13.78379 -1.398832  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
##      1  
##  
## Coefficients:  
##              Value Std.Error   t-value p-value  
## (Intercept) -0.9301408 1.2460336 -0.7464814 0.4744  
## fmSingle-prey 1.8072006 1.3331935 1.3555427 0.2083  
## x_mean       0.6039625 0.2726807 2.2149077 0.0540  
## fmSingle-prey:x_mean -0.7706887 0.3008796 -2.5614521 0.0306  
##  
## Correlation:  
##              (Intr) fmSng- x_mean  
## fmSingle-prey -0.935  
## x_mean       -0.966 0.903  
## fmSingle-prey:x_mean 0.875 -0.954 -0.906  
##  
## Standardized residuals:  
##      Min      Q1      Med      Q3      Max  
## -1.3907385 -0.4392539 0.5816429 0.9134114 1.2865004  
##  
## Residual standard error: 0.370682  
## Degrees of freedom: 13 total; 9 residual  
  
m.pgls.param <- as.data.frame(t(summary(m.pgls.nlme)$tTable[,1])) %>%  
  mutate(intercept.rorq = `(Intercept)`,  
         intercept.od = `(Intercept)` + `fmSingle-prey`,  
         slope.rorq = `x_mean`, slope.od = `x_mean` + `fmSingle-prey:x_mean`)  
m.pgls.param <- m.pgls.param[5:8]
```

6.4.6.1 Model diagnostics

6.4.6.1.1 QQ-plot and Residuals vs fitted plot

```
par(mfrow = c(1,2))  
qqPlot(residuals(m.pgls.nlme), id = FALSE)  
plot(fitted(m.pgls.nlme), residuals(m.pgls.nlme))  
abline(0,0)
```



6.5 Estimate confidence intervals by bootstrapping

6.5.1 Bootstrap and compute percentile confidence intervals

```
d_sub <- filter(d_full, MR.exponent==.61)
index <- d_sub %>% group_by(Spec) %>% summarize(ix=length(y))
index # number of prey categories for each species
```

```
## # A tibble: 13 x 2
##   Spec                ix
##   <fct>                <int>
## 1 Balaenoptera_bonaerensis    5
## 2 Balaenoptera_musculus      7
## 3 Balaenoptera_physalus      7
## 4 Berardius_bairdii         19
## 5 Globicephala_macrorhynchus 12
## 6 Globicephala_melas        12
## 7 Grampus_griseus           5
## 8 Megaptera_novaeangliae     8
## 9 Mesoplodon_densirostris    3
## 10 Orcinus_orca              12
## 11 Phocoena_phocoena         5
## 12 Physeter_macrocephalus    18
## 13 Ziphius_cavirostris      16
```

```

smydata.orig <- smydata
y_mean <- by(d_sub, d_sub$Spec, with, weighted.mean(y, Percent))
spec <- unique(d_sub$Spec)
spec <- spec[match(spec,smydata$species)]

runpGls <- function(smydata,smytree){
  out <- tryCatch(
    {
      model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
        corPagel(lambda.est, phy = smytree, fixed = FALSE),
        method = "REML")
      as.numeric(model.pgls$modelStruct[1])
    },
    error=function(cond) {
      return(NA)
    }
  )
}

a.ols <- matrix(nrow=10000,ncol=4)
a.pgls <- matrix(nrow=10000,ncol=4)
b <- matrix(nrow=10000,ncol=length(spec))
boot.lambdas <- rep(NA,10000)
for(i in 1:10000){
  for(sp in 1:length(spec)){
    ix <- sample(1:index$ix[index$Spec==spec[sp]], replace = T)
    y_mean[sp] <- sum(d_sub[d_sub$Spec==spec[sp],"y"][ix]*
      d_sub[d_sub$Spec==spec[sp],"Percent"][ix])/
      sum(d_sub[d_sub$Spec==spec[sp],"Percent"][ix])
  }
  smydata$y_mean <- y_mean

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
  myout <- runpGls(smydata,smytree)
  boot.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(lambda.est, phy = smytree, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(l.est, phy = smytree, fixed = TRUE), method = "REML")
  }

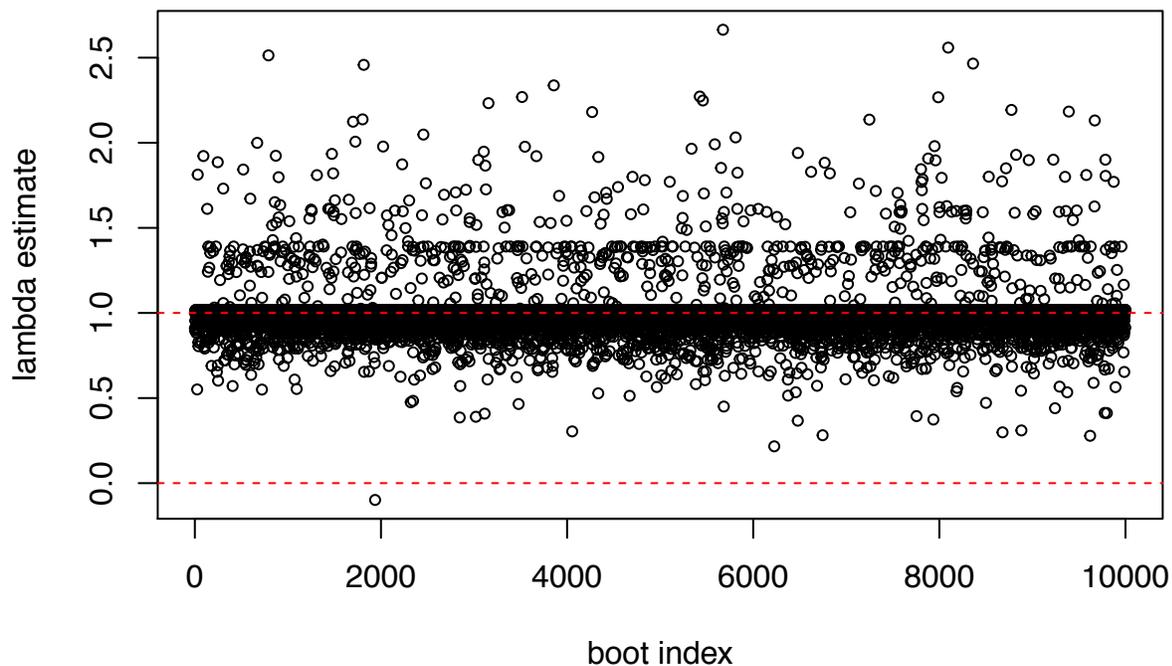
  a.ols[i,] <- c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4])
  a.pgls[i,] <- c(coef(model.pgls)[1],coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],coef(model.pgls)[3]+coef(model.pgls)[4])
  b[i,] <- predict(model.ols)
}

```

```
# number of pGLS models, where lambda could not be estimated ==> used original value:
sum(is.na(boot.lambdas))
```

```
## [1] 234
```

```
plot(boot.lambdas, cex=.7, xlab="boot index", ylab="lambda estimate")
abline(h=0,lty="dashed",col="red")
abline(h=1,lty="dashed",col="red")
```



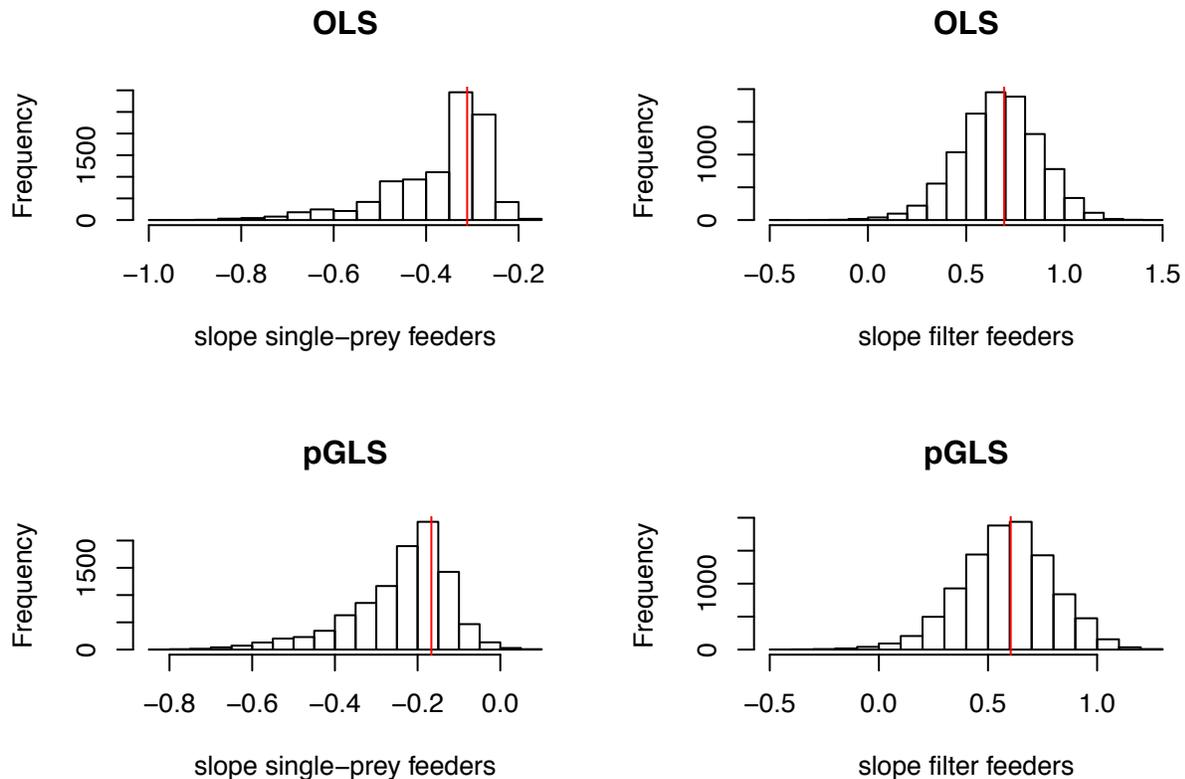
```
preds <- apply(b, 2, quantile, c(0.025, 0.975))
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_61_bootstrap_b.rds")
saveRDS(b,fil)
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_61_bootstrap_preds.rds")
saveRDS(preds,fil)

df.boot.ols <- data.frame(cbind(t(m.ols.param),t(t(apply(a.ols, 2, mean))),
                             t(apply(a.ols, 2, quantile, c(0.025, 0.975)))))
names(df.boot.ols) <- c("obs","bootest","lowerCI","upperCI")
df.boot.pgls <- data.frame(cbind(t(m.pgls.param),t(t(apply(a.pgls, 2, mean))),
                             t(apply(a.pgls, 2, quantile, c(0.025, 0.975)))))
names(df.boot.pgls) <- c("obs","bootest","lowerCI","upperCI")

par(mfrow=c(2,2))
hist(a.ols[,4], xlab="slope single-prey feeders", main="OLS")
abline(v=m.ols.param[4], col="red")
```

```
hist(a.ols[,3], xlab="slope filter feeders", main="OLS")
abline(v=m.ols.param[3], col="red")
```

```
hist(a.pgl[,4], xlab="slope single-prey feeders", main="pGLS")
abline(v=m.pgl.param[4], col="red")
hist(a.pgl[,3], xlab="slope filter feeders", main="pGLS")
abline(v=m.pgl.param[3], col="red")
```



6.5.2 Compute BCa (bias-corrected and accelerated) confidence intervals

```
smydata <- smydata.orig

# compute bias-correction factor from the proportion of bootstrap estimates
# that are less than the observed estimate

bootBC <- function(bootEst, Est){
  B <- ncol(bootEst)*nrow(bootEst) # number of bootstrap samples
  propLess <- sum(bootEst < Est)/B # proportion of replicates less than observed stat
  z0 <- qnorm(propLess) # bias correction
  return(z0)
}

z0.ols <- numeric()
for (i in 1:ncol(a.ols)){
  z0.ols[i] <- bootBC(t(t(a.ols[,i])),as.numeric(m.ols.param[i]))
}
```

```

}

z0.pgls <- numeric()
for (i in 1:ncol(a.pgls)){
  z0.pgls[i] <- bootBC(t(t(a.pgls[,i])),as.numeric(m.pgls.param[i]))
}

# compute acceleration factor, which is related to the skewness of bootstrap estimates.
# Use jackknife replicates to estimate.

jStat.ols <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jStat.pgls <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jack.lambdas <- rep(NA,nrow(smydata))
for (i in 1:nrow(smydata)) {
  d_sub <- subset(d_full, Spec==smydata$species[i] & MR.exponent==.61)
  y_mean.j <- numeric()
  for(j in 1:nrow(d_sub)){
    d_sub.j <- d_sub[-j,]
    y_mean.j[j] <- sum(d_sub.j$y*d_sub.j$Percent)/sum(d_sub.j$Percent)
  }
  smydata.j <- smydata
  smydata.j$y_mean[i] <- mean(y_mean.j)
  pruned.tree <- drop.tip(smytree,smytree$tip.label[-match(smydata.j$species,
    smytree$tip.label)])

  smytree.j <- pruned.tree
  smydata.j <- smydata.j[match(smytree.j$tip.label,rownames(smydata.j)),]

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata.j, method = "REML")

  myout <- runpGls(smydata.j,smytree.j)
  jack.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(lambda.est, phy = smytree.j, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(l.est, phy = smytree.j, fixed = TRUE), method = "REML")
  }

  jStat.ols[,i] <- as.numeric(c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4]))
  jStat.pgls[,i] <- as.numeric(c(coef(model.pgls)[1],
    coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],
    coef(model.pgls)[3]+coef(model.pgls)[4]))
}

jackEst.ols <- t(t(apply(jStat.ols, 1, mean))) # jackknife estimate
jackEst.pgls <- t(t(apply(jStat.pgls, 1, mean))) # jackknife estimate

```

```

jack.lambdas # lambdas of the jackknifed models

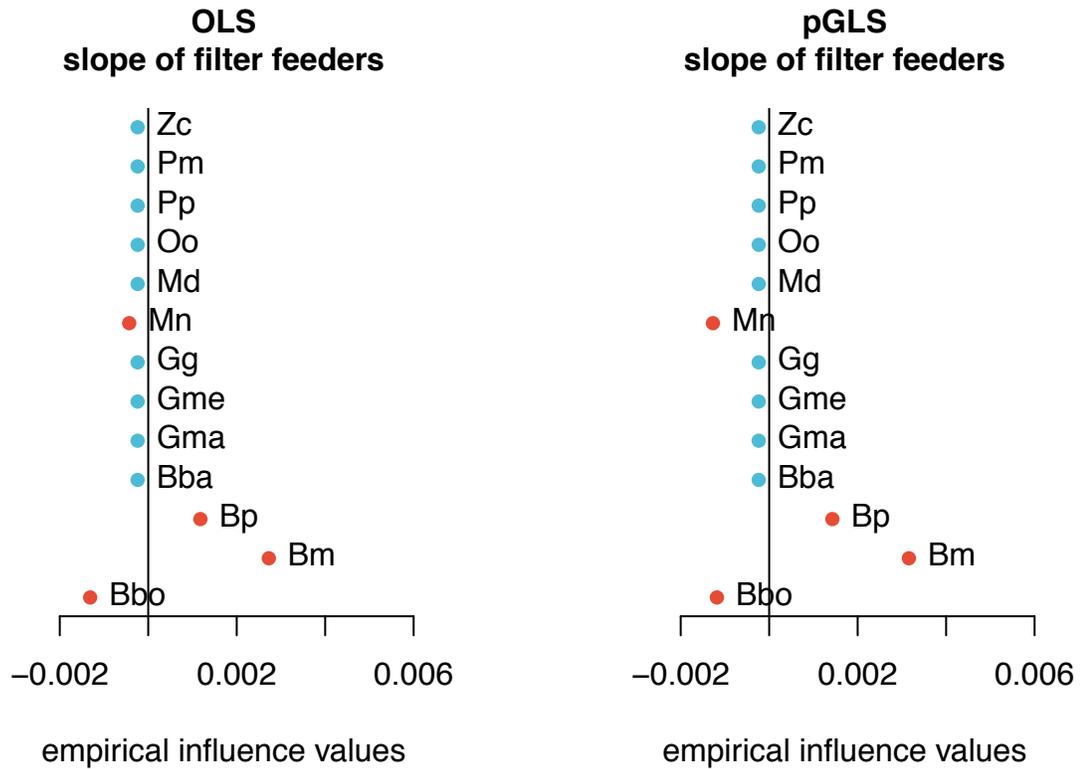
## [1] 1.005427 1.005473 1.005366 1.005432 1.005710 1.005148 1.005370
## [8] 1.005379 1.005269 1.005406 1.004647 1.005424 1.005416

num.ols <- numeric(); den.ols <- numeric(); ahat.ols <- numeric()
num.pgls <- numeric(); den.pgls <- numeric(); ahat.pgls <- numeric()
for (i in 1:nrow(jStat.ols)){
  num.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^3 )
  den.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^2 )
  ahat.ols[i] <- num.ols[i]/(6*den.ols[i]^(3/2)) # ahat based on jackknife
  num.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^3 )
  den.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^2 )
  ahat.pgls[i] <- num.pgls[i]/(6*den.pgls[i]^(3/2)) # ahat based on jackknife
}

# influential species:
par(mfrow=c(1,2))
plot(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("OLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("pGLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```

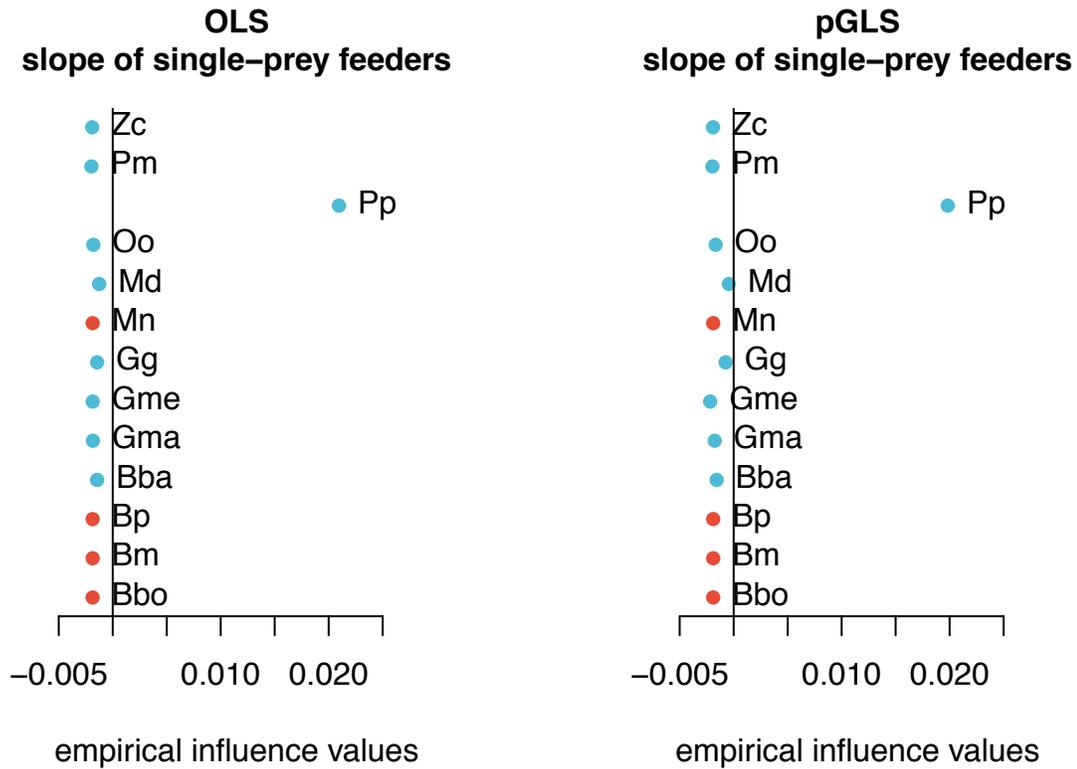


```

par(mfrow=c(1,2))
plot(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                   round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3))
title(paste0("OLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                   round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3))
title(paste0("pGLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```



```
# adjust quantiles for 100*(1-alpha)% bootstrap BCa interval
```

```
alpha <- 0.05
zL.ols <- z0.ols + qnorm(alpha/2)
alpha1.ols <- pnorm(z0.ols + zL.ols / (1-ahat.ols*zL.ols))
zU.ols <- z0.ols + qnorm(1-alpha/2)
alpha2.ols <- pnorm(z0.ols + zU.ols / (1-ahat.ols*zU.ols))

zL.pgls <- z0.pgls + qnorm(alpha/2)
alpha1.pgls <- pnorm(z0.pgls + zL.pgls / (1-ahat.pgls*zL.pgls))
zU.pgls <- z0.pgls + qnorm(1-alpha/2)
alpha2.pgls <- pnorm(z0.pgls + zU.pgls / (1-ahat.pgls*zU.pgls))
```

```
cbind((alpha1.ols*100), (alpha2.ols*100)) # new quantiles OLS:
```

```
##           [,1]      [,2]
## [1,]  0.54081431  93.76842
## [2,]  0.00857251  83.98231
## [3,]  6.69124330  99.54614
## [4,] 16.62300436  99.99301
```

```
cbind((alpha1.pgls*100), (alpha2.pgls*100)) # new quantiles pGLS:
```

```
##           [,1]      [,2]
## [1,] 8.176548e-01  94.87697
## [2,] 1.635101e-04  70.63279
## [3,] 5.520553e+00  99.31704
```

```

## [4,] 2.906833e+01 99.99983
CI.ols <- matrix(nrow = ncol(a.ols), ncol=2)
for (i in 1:ncol(a.ols)){
  CI.ols[i,] <- quantile(a.ols[,i], c(alpha1.ols[i], alpha2.ols[i])) # BCa interval
}
df.boot.ols$lowerCIbca <- CI.ols[,1]
df.boot.ols$upperCIbca <- CI.ols[,2]

CI.pgls <- matrix(nrow = ncol(a.pgls), ncol=2)
for (i in 1:ncol(a.pgls)){
  CI.pgls[i,] <- quantile(a.pgls[,i], c(alpha1.pgls[i], alpha2.pgls[i])) # BCa interval
}
df.boot.pgls$lowerCIbca <- CI.pgls[,1]
df.boot.pgls$upperCIbca <- CI.pgls[,2]

```

6.5.3 Plot OLS model

```

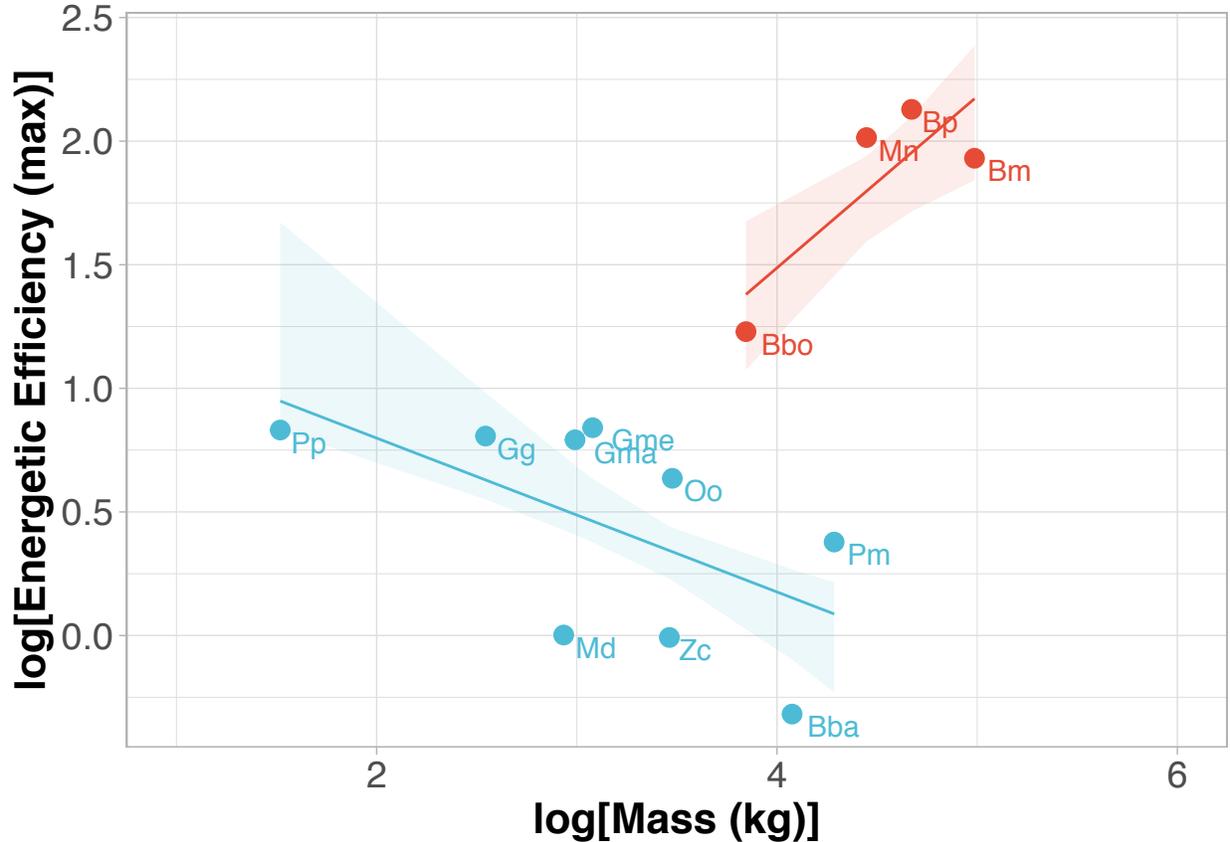
smydata <- smydata.orig

ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit,
                                     y_min = preds[1,], y_max = preds[2,]))

fig_ols <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
           linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
           linetype = 1) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Rorqual"), fill = "#E64B35FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Odontocete"), fill = "#4DBBD5FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
           aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
           aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_ols

```



6.5.3.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCI"],
  lowerCI.od = df.boot.ols["slope.od","lowerCI"],
  upperCI.od = df.boot.ols["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.ols["slope.od","lowerCIbca"],
  upperCI.od = df.boot.ols["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.ols[,3],
  odontocete_slope=a.ols[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF", color = NA, alpha = 0.3)) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF", color = NA, alpha = 0.3)) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

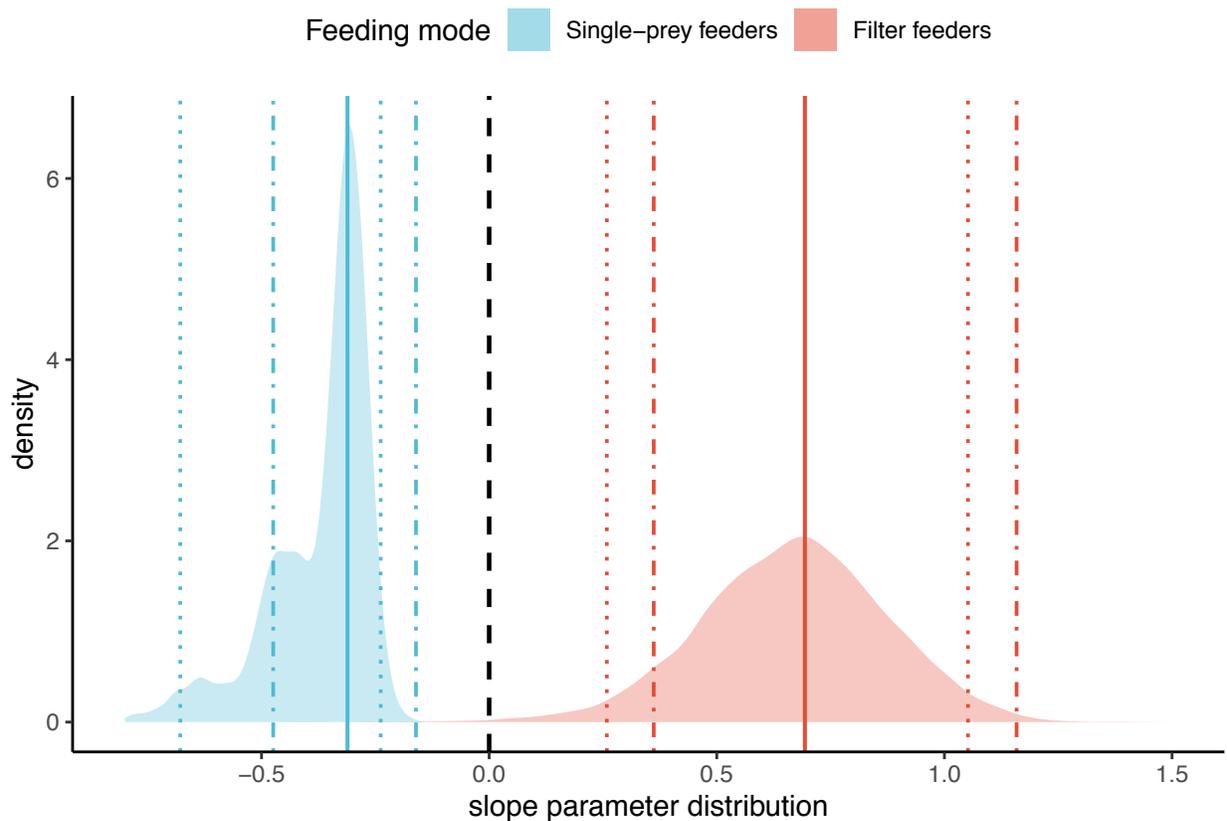
```

```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 41 rows containing non-finite values (stat_density).



```

rn <- rownames(df.boot.ols)
rownames(df.boot.ols) <- c("intercept filter", "intercept single-prey",

```

```

      "slope filter", "slope single-prey")
knitr::kable(df.boot.ols,
             caption = "OLS 95% Bootstrap Pctl and BCa CI",
             format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
  kable_styling(latex_options = "hold_position")

```

Table 9: OLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-1.2869	-1.1902	-2.9310	0.6511	-3.3437	0.2135
intercept single-prey	1.4210	1.6281	1.2026	2.6999	0.9334	1.9962
slope filter	0.6936	0.6678	0.2585	1.0518	0.3615	1.1586
slope single-prey	-0.3112	-0.3711	-0.6786	-0.2383	-0.4743	-0.1608

```
rownames(df.boot.ols) <- rn
```

6.5.4 Plot pGLS model

```

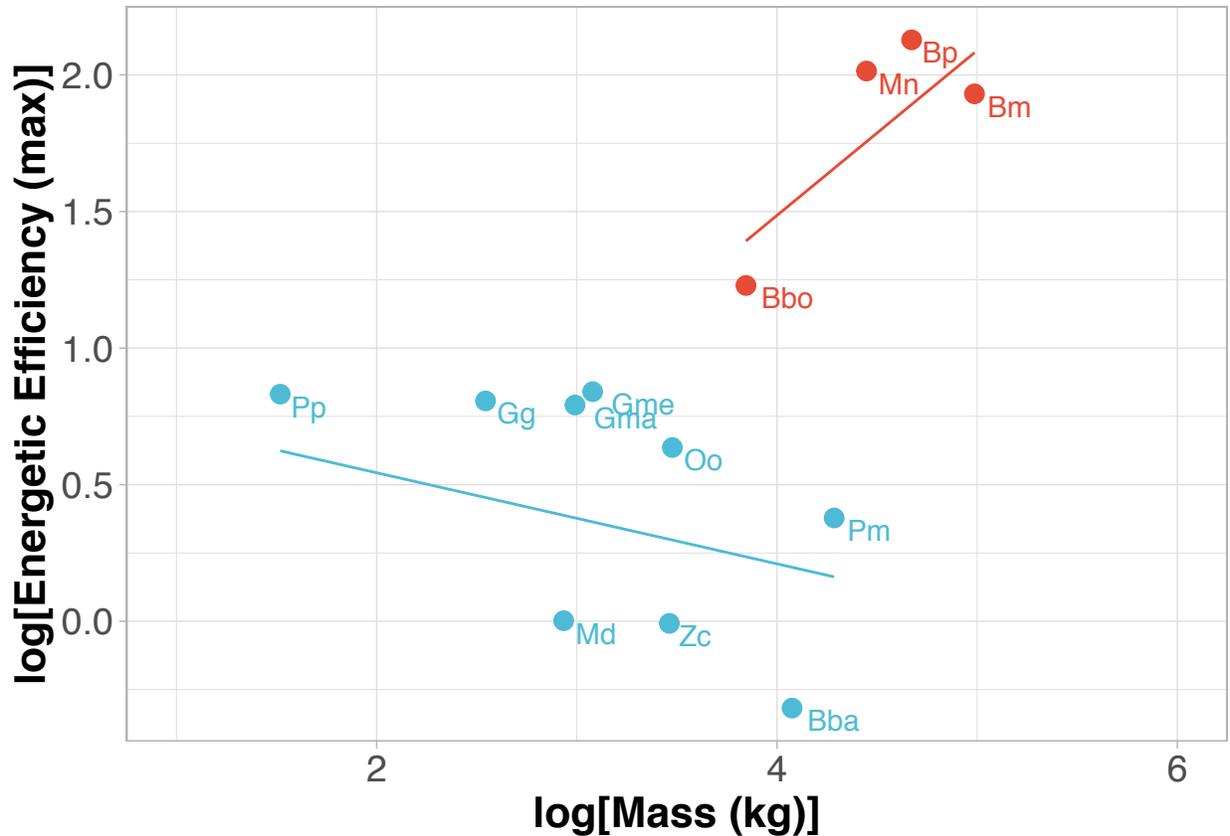
pgls.fit <- predict(m.pgls.nlme)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))

fig_pgls <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
            linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
            linetype = 1) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +

  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_pgls

```



6.5.4.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.pglis["slope.rorq","obs"],
  slope.od = df.boot.pglis["slope.od","obs"],
  lowerCI.rorq = df.boot.pglis["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.pglis["slope.rorq","upperCI"],
  lowerCI.od = df.boot.pglis["slope.od","lowerCI"],
  upperCI.od = df.boot.pglis["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.pglis["slope.rorq","obs"],
  slope.od = df.boot.pglis["slope.od","obs"],
  lowerCI.rorq = df.boot.pglis["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.pglis["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.pglis["slope.od","lowerCIbca"],
  upperCI.od = df.boot.pglis["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.pglis[,3],
  odontocete_slope=a.pglis[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF", color = NA, alpha = 0.3)) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF", color = NA, alpha = 0.3)) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

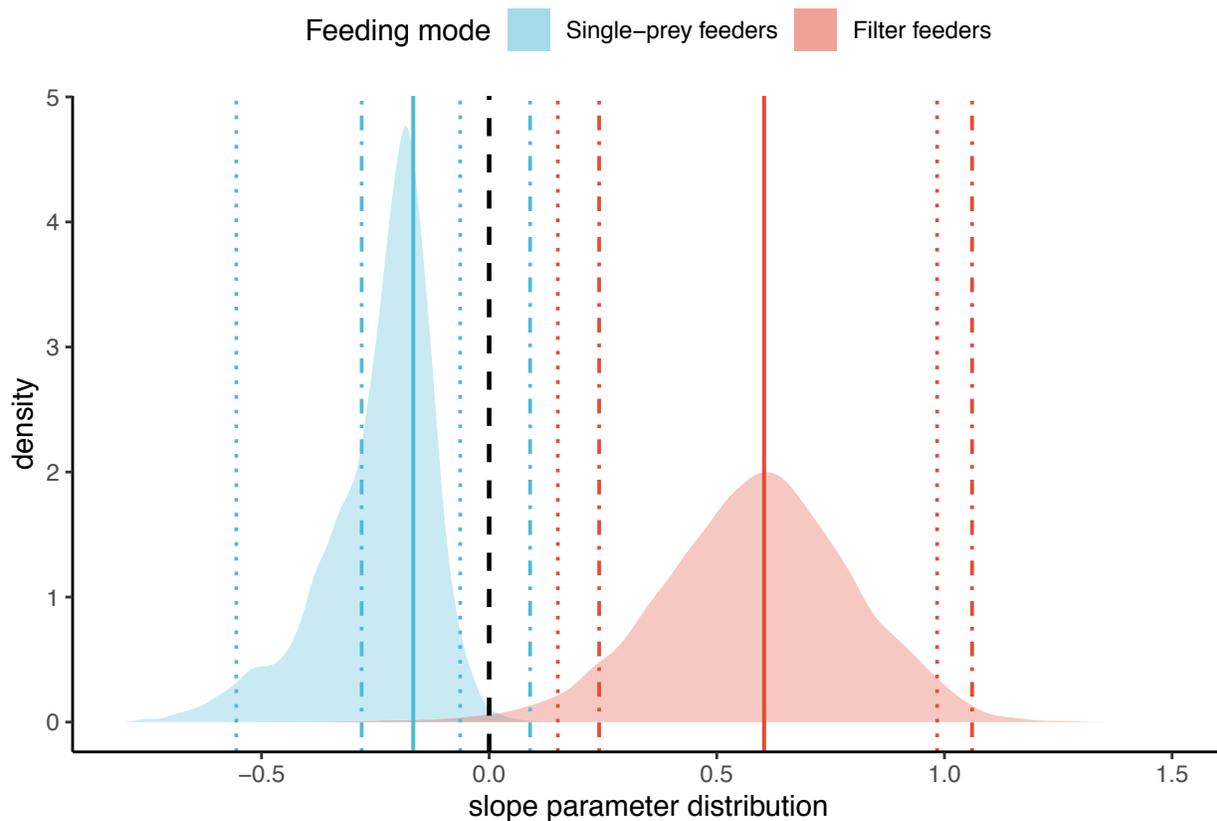
```

```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 1 rows containing non-finite values (stat_density).



```

rn <- rownames(df.boot.pgl)
rownames(df.boot.pgl) <- c("intercept filter", "intercept single-prey",

```

```

      "slope filter", "slope single-prey")
knitr::kable(df.boot.pgls,
  caption = "pGLS 95% Bootstrap Pctl and BCa CI",
  format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
  kable_styling(latex_options = "hold_position")

```

Table 10: pGLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-0.9301	-0.8682	-2.6766	1.0638	-2.9702	0.7068
intercept single-prey	0.8771	1.1512	0.5346	2.3277	-0.0082	1.2758
slope filter	0.6040	0.5870	0.1509	0.9839	0.2416	1.0608
slope single-prey	-0.1667	-0.2415	-0.5554	-0.0637	-0.2801	0.0900

```
rownames(df.boot.pgls) <- rn
```

6.6 Extract summary statistics

```

specify_decimal <- function(x, k) trimws(format(round(x, k), nsmall = k))

res.df.ols <- m.ols$dimN - m.ols$dimP

res.df.pgls <- m.pgls.nlme$dimN - m.pgls.nlme$dimP

intercepts.od.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCIbca"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCIbca"], 4),
  ")"))

intercepts.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCIbca"],
  4), ")"),
  paste0(specify_decimal(df.boot.ols["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.rorq", "upperCI"], 4),
  ")"))

```

```

paste0(specify_decimal(df.boot.ols["intercept.rorq","obs"],4),
"(", specify_decimal(df.boot.ols["intercept.rorq","lowerCIbca"],4),
" - ", specify_decimal(df.boot.ols["intercept.rorq","upperCIbca"],4),
")"))

slopes.od.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.od","obs"],4),"(",
specify_decimal(df.boot.pgl["slope.od","lowerCI"],4)," - ",
specify_decimal(df.boot.pgl["slope.od","upperCI"],4),")"),
paste0(specify_decimal(df.boot.pgl["slope.od","obs"],4),"(",
specify_decimal(df.boot.pgl["slope.od","lowerCIbca"],4)," - ",
specify_decimal(df.boot.pgl["slope.od","upperCIbca"],4),")"),
paste0(specify_decimal(df.boot.ols["slope.od","obs"],4),"(",
specify_decimal(df.boot.ols["slope.od","lowerCI"],4)," - ",
specify_decimal(df.boot.ols["slope.od","upperCI"],4),")"),
paste0(specify_decimal(df.boot.ols["slope.od","obs"],4),"(",
specify_decimal(df.boot.ols["slope.od","lowerCIbca"],4)," - ",
specify_decimal(df.boot.ols["slope.od","upperCIbca"],4),")"))

slopes.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.rorq","obs"],4),"(",
specify_decimal(df.boot.pgl["slope.rorq","lowerCI"],4)," - ",
specify_decimal(df.boot.pgl["slope.rorq","upperCI"],4),")"),
paste0(specify_decimal(df.boot.pgl["slope.rorq","obs"],4),"(",
specify_decimal(df.boot.pgl["slope.rorq","lowerCIbca"],4),
" - ", specify_decimal(df.boot.pgl["slope.rorq","upperCIbca"],
4),")"),
paste0(specify_decimal(df.boot.ols["slope.rorq","obs"],4),"(",
specify_decimal(df.boot.ols["slope.rorq","lowerCI"],4)," - ",
specify_decimal(df.boot.ols["slope.rorq","upperCI"],4),")"),
paste0(specify_decimal(df.boot.ols["slope.rorq","obs"],4),"(",
specify_decimal(df.boot.ols["slope.rorq","lowerCIbca"],4),
" - ", specify_decimal(df.boot.ols["slope.rorq","upperCIbca"],
4),")"))

a.od.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.od","obs"]),4),"(",
specify_decimal(10^(df.boot.pgl["intercept.od","lowerCI"]),4),
" - ", specify_decimal(10^(df.boot.pgl["intercept.od","upperCI"]),4),
")"),
paste0(specify_decimal(10^(df.boot.pgl["intercept.od","obs"]),4),"(",
specify_decimal(10^(df.boot.pgl["intercept.od","lowerCIbca"]),4),
" - ", specify_decimal(10^(df.boot.pgl["intercept.od","upperCIbca"]),4),
")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od","obs"]),4),"(",
specify_decimal(10^(df.boot.ols["intercept.od","lowerCI"]),4)," - ",
specify_decimal(10^(df.boot.ols["intercept.od","upperCI"]),4),")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od","obs"]),4),"(",
specify_decimal(10^(df.boot.ols["intercept.od","lowerCIbca"]),4),
" - ", specify_decimal(10^(df.boot.ols["intercept.od","upperCIbca"])),
4),")"))

a.rorq.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq","obs"]),4),
"(", specify_decimal(10^(df.boot.pgl["intercept.rorq","lowerCI"]),5),
" - ", specify_decimal(10^(df.boot.pgl["intercept.rorq","upperCI"]),4),")"),
paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq","obs"]),4),
"(", specify_decimal(10^(df.boot.pgl["intercept.rorq","lowerCIbca"]),5),
" - ", specify_decimal(10^(df.boot.pgl["intercept.rorq","upperCIbca"]),4),

```

Table 11: Model summary statistics

	<i>Filter feeders</i>		<i>Single-prey feeders</i>		RSE	tot.df	res.df
	slope*	intercept	slope	intercept			
pGLS	0.6040 (0.1509 - 0.9839)	-0.9301 (-2.6766 - 1.0638)	-0.1667 (-0.5554 - -0.0637)	0.8771 (0.5346 - 2.3277)	0.3707		
	0.6040 (0.2416 - 1.0608)	-0.9301 (-2.9702 - 0.7068)	-0.1667 (-0.2801 - 0.0900)	0.8771 (-0.0082 - 1.2758)	0.3707		
OLS	0.6936 (0.2585 - 1.0518)	-1.2869 (-2.9310 - 0.6511)	-0.3112 (-0.6786 - -0.2383)	1.4210 (1.2026 - 2.6999)	0.3668	13	9
	0.6936 (0.3615 - 1.1586)	-1.2869 (-3.3437 - 0.2135)	-0.3112 (-0.4743 - -0.1608)	1.4210 (0.9334 - 1.9962)	0.3668		

Note:

* Throughout the table, values in brackets represent 95% confidence intervals: percentile in shaded rows, BCa in non-shaded rows.

```

    ")),
    paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
    " (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCI"]),5),
    " - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCI"]),4),")"),
    paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
    " (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCIbca"]),5),
    " - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCIbca"]),4),
    ")))

RSE <- rbind(specify_decimal(t(t(rep(as.numeric(m.pgl$nlme$sigma),2))),4),
             specify_decimal(t(t(rep(as.numeric(m.ols$sigma),2))),4))
df <- cbind(t(t(c(rep(m.pgl$nlme$dim$N,2), rep(m.ols$dim$N,2)))),
            t(t(c(rep(res.df.pgl,2), rep(res.df.ols,2)))))
models <- rbind(t(t(rep("pGLS",2))),t(t(rep("OLS",2)))

outputs <- cbind(models, slopes.rorq.ci, intercepts.rorq.ci, slopes.od.ci,
                 intercepts.od.ci, RSE, df)
df.outputs <- data.frame(outputs, check.rows = TRUE, check.names = TRUE)
names(df.outputs) <- c("","slope","intercept","slope","intercept","RSE", "tot.df","res.df")
names(df.outputs)[2] <- paste0(names(df.outputs)[2],
                               footnote_marker_symbol(1))

knitr::kable(df.outputs,
             caption = "Model summary statistics",
             format = "latex", booktabs = TRUE, escape = FALSE) %>%
  kable_styling(latex_options = "scale_down") %>%
  row_spec(0, bold = T) %>%
  row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
  column_spec(c(1,(ncol(df.outputs)-1):ncol(df.outputs))-1,
              background = "white") %>%
  column_spec(1, bold = T) %>%
  collapse_rows(columns = c(1,(ncol(df.outputs)-1):ncol(df.outputs))) %>%
  add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2,
                    " " = 3), bold = T, italic = T) %>%
  footnote(general = "", general_title = "Note:",
           symbol = paste0("Throughout the table, values in brackets",
                           " represent 95% confidence intervals: ",
                           "percentile in shaded rows, BCa in non-shaded rows."),
           symbol_title = "", title_format = "italic",
           footnote_as_chunk = T)

alloout <- cbind(models, a.rorq.ci, slopes.rorq.ci, a.od.ci, slopes.od.ci)
df.allo <- data.frame(alloout, check.rows = TRUE, check.names = TRUE)
names(df.allo) <- c("","a","b","a","b")

```

Table 12: Transformed to allometric equations

	<i>Filter feeders</i>		<i>Single-prey feeders</i>	
	<i>a</i> *	<i>b</i>	<i>a</i>	<i>b</i>
pGLS	0.1175 (0.00211 - 11.5815)	0.6040 (0.1509 - 0.9839)	7.5346 (3.4243 - 212.6773)	-0.1667 (-0.5554 - -0.0637)
	0.1175 (0.00107 - 5.0906)	0.6040 (0.2416 - 1.0608)	7.5346 (0.9812 - 18.8693)	-0.1667 (-0.2801 - 0.0900)
OLS	0.0517 (0.00117 - 4.4780)	0.6936 (0.2585 - 1.0518)	26.3638 (15.9428 - 501.0831)	-0.3112 (-0.6786 - -0.2383)
	0.0517 (0.00045 - 1.6349)	0.6936 (0.3615 - 1.1586)	26.3638 (8.5785 - 99.1244)	-0.3112 (-0.4743 - -0.1608)

* Throughout the table, values in brackets represent 95% confidence intervals.: percentile in shaded rows, BCa in non-shaded rows.

```
names(df.allo)[2] <- paste0(names(df.allo)[2], footnote_marker_symbol(1))
knitr::kable(df.allo,
  caption = "Transformed to allometric equations",
  format = "latex", booktabs = TRUE, escape = FALSE) %>%
  kable_styling(latex_options = "scale_down") %>%
  row_spec(0, bold = T) %>%
  row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
  column_spec(1, bold = T) %>%
  collapse_rows(columns = 1) %>%
  add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2),
    bold = T, italic = T) %>%
  footnote(symbol = paste0("Throughout the table, values in brackets",
    " represent 95% confidence intervals.: ",
    "percentile in shaded rows, BCa in non-shaded rows."),
    symbol_title = "", threeparttable = TRUE, footnote_as_chunk = T)
```

6.7 Plot best models (OLS - dashed, PGLS - solid)

```
pgls.fit <- predict(m.pgls.nlme)
ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))

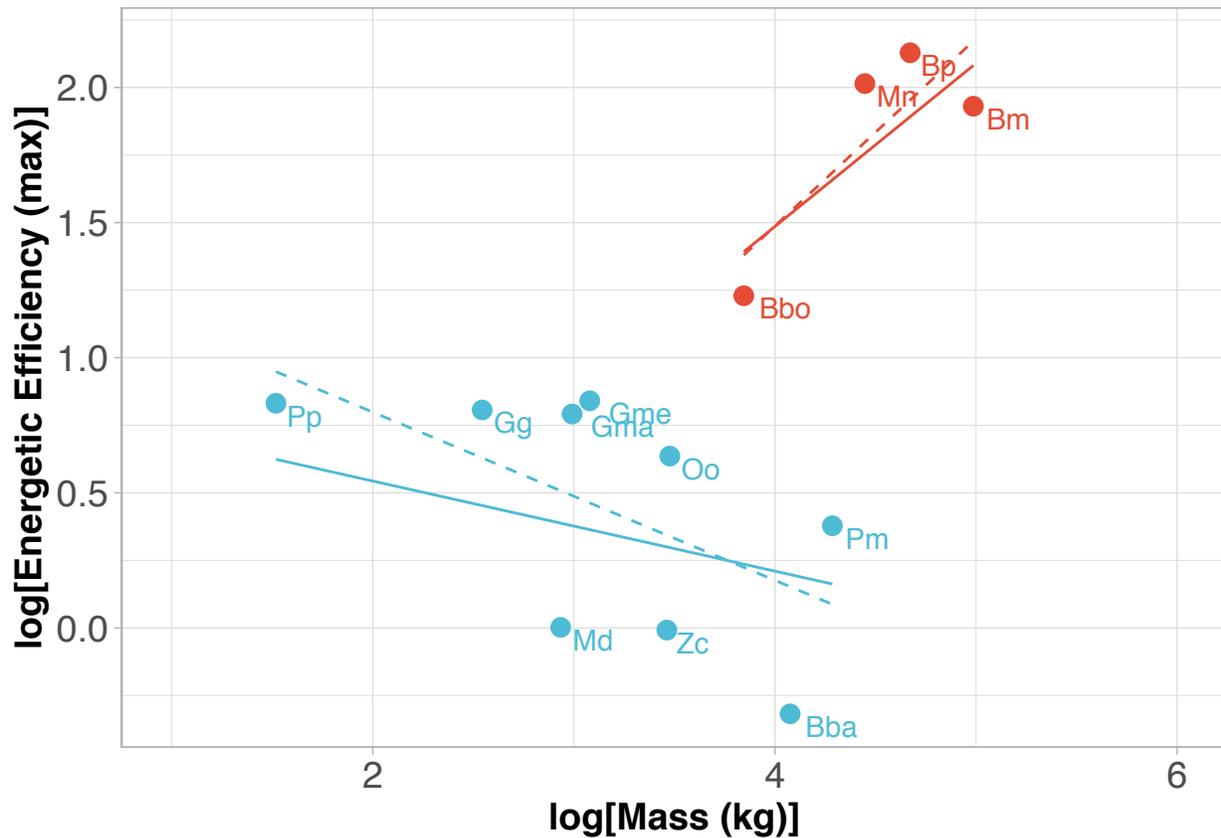
fig_4.61 <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe2, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 2) +
  geom_line(data = dplyr::filter(predframe2, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 2) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  xlim(1,6) +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 14,
```

```

labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    aes(label = abbreviation, hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

```

fig_4.61



6.7.1 Construct output table

```

df.out <- smydata[,c("species", "fm", "x_mean", "y_mean")]
df.out$fitted_ols <- fitted(m.ols)
df.out$fitted_pgls <- fitted(m.pgls.nlme)
rownames(df.out) <- NULL
kable(df.out,
  caption = "Model outputs",
  format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")

```

6.8 Quick clean up

```

m.61.pgls.nlme <- m.pgls.nlme
df.61.outputs <- df.outputs
m.61.ols <- m.ols

```

Table 13: Model outputs

species	fm	x_mean	y_mean	fitted_ols	fitted_pgl
Balaenoptera_bonaerensis	Filter	3.8451	1.2293	1.3799	1.3922
Balaenoptera_musculus	Filter	4.9868	1.9307	2.1718	2.0817
Balaenoptera_physalus	Filter	4.6725	2.1286	1.9538	1.8919
Berardius_bairdii	Single-prey	4.0755	-0.3180	0.1527	0.1976
Globicephala_macrorhynchus	Single-prey	2.9912	0.7918	0.4901	0.3783
Globicephala_melas	Single-prey	3.0792	0.8406	0.4627	0.3637
Grampus_griseus	Single-prey	2.5441	0.8068	0.6293	0.4529
Megaptera_novaeangliae	Filter	4.4472	2.0144	1.7975	1.7558
Mesoplodon_densirostris	Single-prey	2.9345	0.0020	0.5078	0.3878
Orcinus_orca	Single-prey	3.4771	0.6359	0.3389	0.2973
Phocoena_phocoena	Single-prey	1.5185	0.8313	0.9484	0.6239
Physeter_macrocephalus	Single-prey	4.2856	0.3781	0.0873	0.1625
Ziphius_cavirostris	Single-prey	3.4624	-0.0079	0.3435	0.2998

```
to.keep <- c(to.keep, "m.61.pgl.nlme", "df.61.outputs", "m.61.ols")
rm(list=setdiff(ls(), to.keep))
```

7 Run model for MR = .68

7.1 Prepare data

7.1.1 Get rid of rows with NAs - subset the data

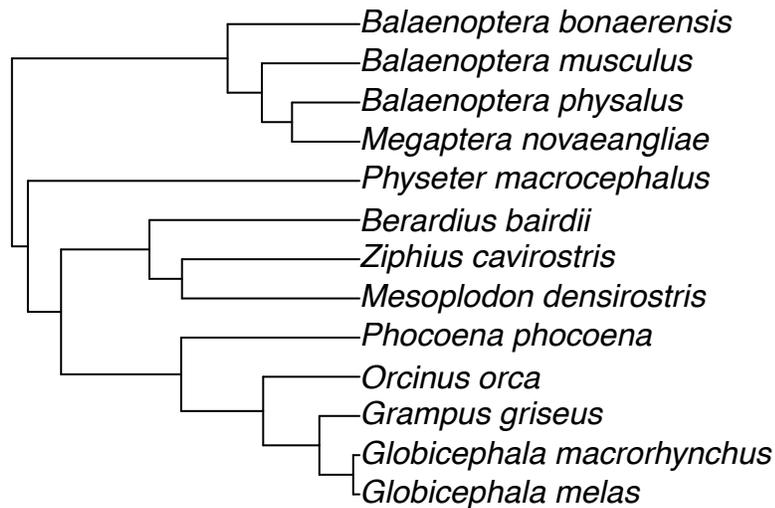
```
smydata <- df.spec
smydata$y_mean <- smydata$wgtMean.68
smydata <- smydata[!is.na(smydata$y_mean),]
smydata <- smydata[!is.na(smydata$x_mean),]
smydata$fm <- factor(smydata$fm)
smydata$Group <- smydata$Group
colnames(smydata)[1] <- "species"
```

7.1.2 Adjust tree - drop species for which data are missing

```
smytree <- drop.tip(mytree, mytree$tip.label[-match(smydata$species, mytree$tip.label)])
plot(smytree)
```

Table 14: Model inputs

species	gr	x_mean	fm	Group	abbreviation	wgtMean.45	wgtMean.61	wgtMean.68	wgtMean.75	y_mean
<i>Balaenoptera bonaerensis</i>	1	3.8451	Filter	Rorqual	Bbo	1.3289	1.2293	1.1386	1.0061	1.1386
<i>Balaenoptera musculus</i>	1	4.9868	Filter	Rorqual	Bm	2.0165	1.9307	1.8293	1.6606	1.8293
<i>Balaenoptera physalus</i>	1	4.6725	Filter	Rorqual	Bp	2.2749	2.1286	2.0246	1.8348	2.0246
<i>Berardius bairdii</i>	5	4.0755	Single-prey	Odontocete	Bba	0.7242	-0.3180	-0.0448	-0.3180	-0.0448
<i>Globicephala macrorhynchus</i>	2	2.9912	Single-prey	Odontocete	Gma	1.0519	0.7918	0.6401	0.4703	0.6401
<i>Globicephala melas</i>	2	3.0792	Single-prey	Odontocete	Gme	1.2829	0.8406	0.6335	0.4257	0.6335
<i>Grampus griseus</i>	2	2.5441	Single-prey	Odontocete	Gg	1.1941	0.8068	0.6328	0.4577	0.6328
<i>Megaptera novaeangliae</i>	1	4.4472	Filter	Rorqual	Mn	2.1839	2.0144	1.8621	1.6563	1.8621
<i>Mesoplodon densirostris</i>	5	2.9345	Single-prey	Odontocete	Md	0.4639	0.0020	-0.2024	-0.4092	-0.2024
<i>Orcinus orca</i>	2	3.4771	Single-prey	Odontocete	Oo	0.7544	0.6359	0.5393	0.4096	0.5393
<i>Phocoena phocoena</i>	3	1.5185	Single-prey	Odontocete	Pp	1.0350	0.8313	0.7372	0.6407	0.7372
<i>Physeter macrocephalus</i>	4	4.2856	Single-prey	Odontocete	Pm	0.7403	0.3781	0.1685	-0.0732	0.1685
<i>Ziphius cavirostris</i>	5	3.4624	Single-prey	Odontocete	Zc	0.4286	-0.0079	-0.2229	-0.4496	-0.2229



7.1.3 Rearrange the row order in smydata to match smytree

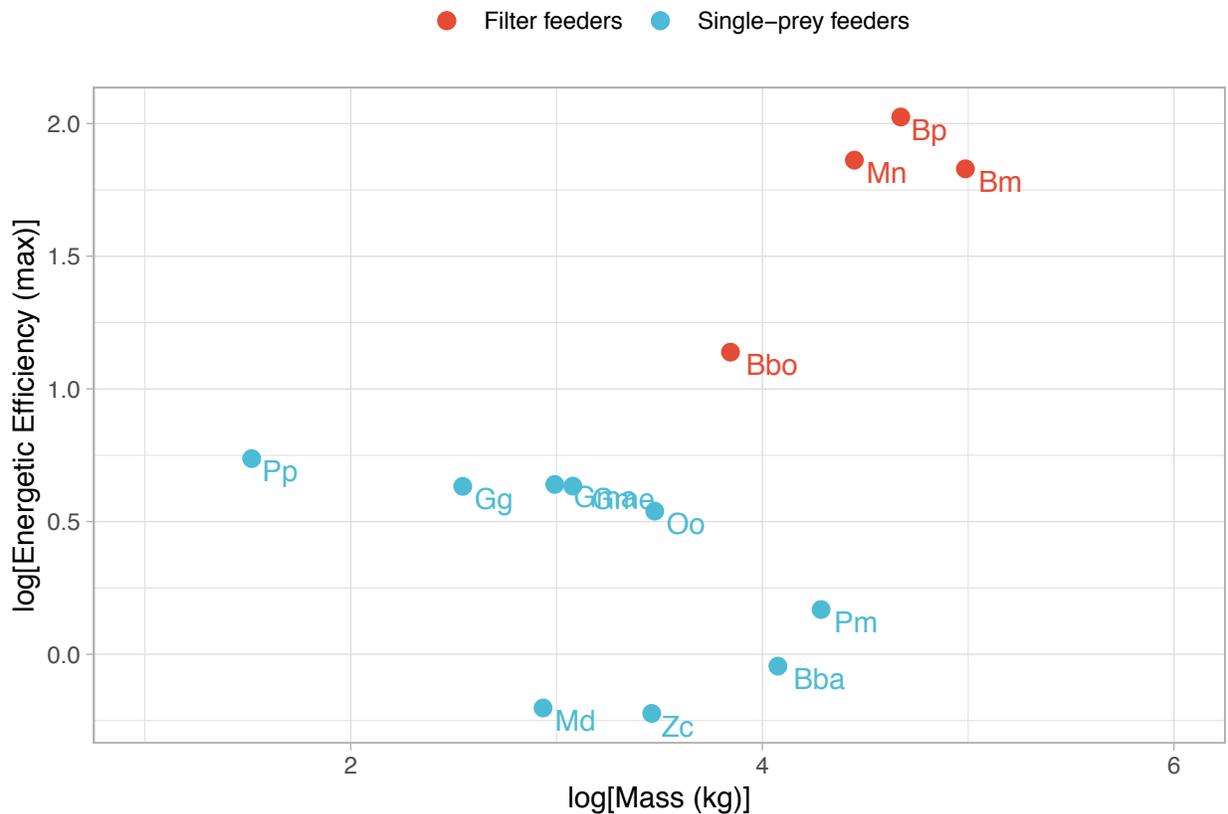
```
smydata <- smydata[match(smytree$tip.label,rownames(smydata)),]
rownames(smydata) <- NULL
kable(smydata,
      caption = "Model inputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")
```

```
rownames(smydata) <- smydata$species
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_68_smydata.rds")
```

```
saveRDS(smydata,fil)
```

7.2 Plot the data

```
ggplot(smydata, aes(x_mean, y = value, color = Group)) +  
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), shape = 16, size = 3,  
    aes(y = y_mean, color = "#4DBBD5FF")) +  
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), shape = 16, size = 3,  
    aes(y = y_mean, color = "#E64B35FF")) +  
  scale_color_manual(name = "",  
    values = c("#E64B35FF", "#4DBBD5FF"),  
    labels = c("Filter feeders", "Single-prey feeders")) +  
  theme_light() + theme(legend.position = "top") +  
  xlim(1,6) +  
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +  
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",  
    aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1) +  
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",  
    aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1)
```



7.3 Run OLS with feeding mode as a categorical predictor

7.3.1 Run OLS and model reduction using ML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "ML")
summary(m.ols)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 13.54421 16.36896 -1.772105
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.3654447 1.7982863 -0.7593033 0.4671
## fmSingle-prey  2.5164840 1.8569964  1.3551367 0.2084
## x_mean        0.6860893 0.3989734  1.7196368 0.1196
## fmSingle-prey:x_mean -0.9496985 0.4237189 -2.2413409 0.0517
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean        -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -2.0910100 -0.4835162  0.5305907  0.6646164  1.0993087
##
## Residual standard error: 0.277309
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.ols)
```

```
## Denom. DF: 9
##      numDF F-value p-value
## (Intercept)  1 65.64109 <.0001
## fm          1 48.41162 0.0001
## x_mean      1  1.34694 0.2757
## fm:x_mean   1  5.02361 0.0517
```

```
m.ols.2 <- update(m.ols, ~ . - fm:x_mean)
anova(m.ols, m.ols.2)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## m.ols      1  5 13.54421 16.36896 -1.772105
## m.ols.2    2  4 17.30994 19.56974 -4.654970 1 vs 2 5.76573 0.0163
```

7.3.1.1 Compare to an intercept-only model

```
m.ols.0 <- gls(y_mean ~ 1, data = smydata, method = "ML")
anova(m.ols, m.ols.0)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## m.ols      1  5 13.54421 16.36896 -1.772105
## m.ols.0    2  2 33.00145 34.13135 -14.500724 1 vs 2 25.45724 <.0001
```

```
m.ols.p <- anova(m.ols, m.ols.0)$`p-value`[2]
```

7.3.2 Estimate final model using REML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
summary(m.ols)
```

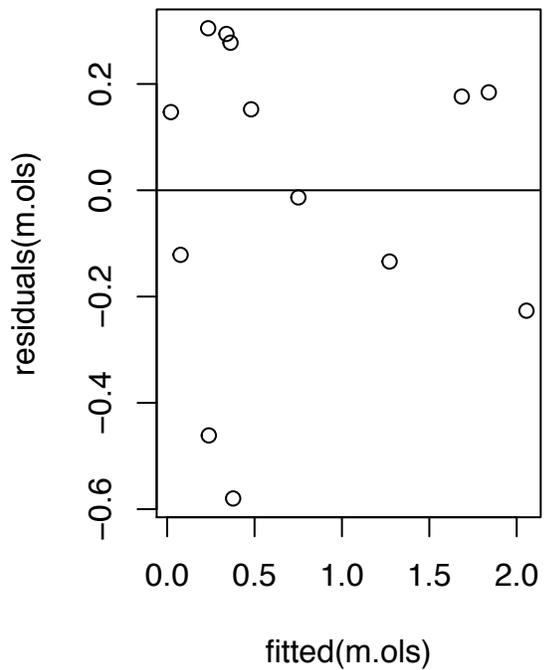
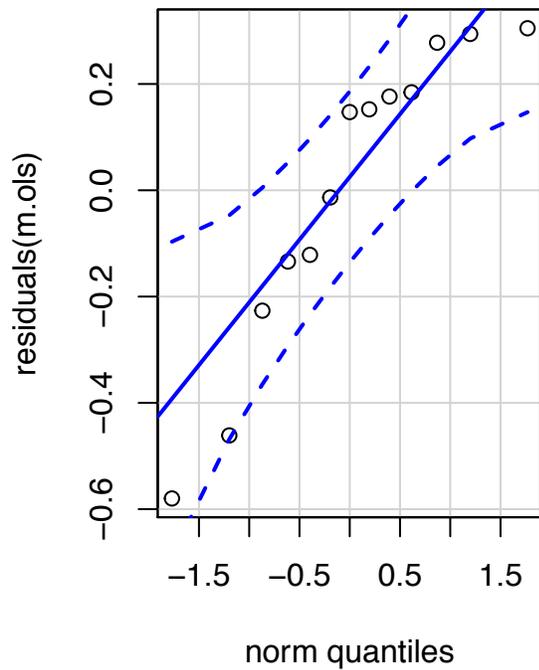
```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 20.68369 21.66981 -5.341845
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.3654447  1.7982863  -0.7593033  0.4671
## fmSingle-prey  2.5164840  1.8569964   1.3551367  0.2084
## x_mean        0.6860893  0.3989734   1.7196368  0.1196
## fmSingle-prey:x_mean -0.9496985  0.4237189  -2.2413409  0.0517
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7398255 -0.4023098  0.4414781  0.5529942  0.9146801
##
## Residual standard error: 0.333284
## Degrees of freedom: 13 total; 9 residual
```

```
m.ols.param <- as.data.frame(t(summary(m.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.ols.param <- m.ols.param[5:8]
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_68_m_ols_param.rds")
saveRDS(m.ols.param,fil)
```

7.3.2.1 Model diagnostics

7.3.2.1.1 QQ-plot and Residuals vs fitted plot

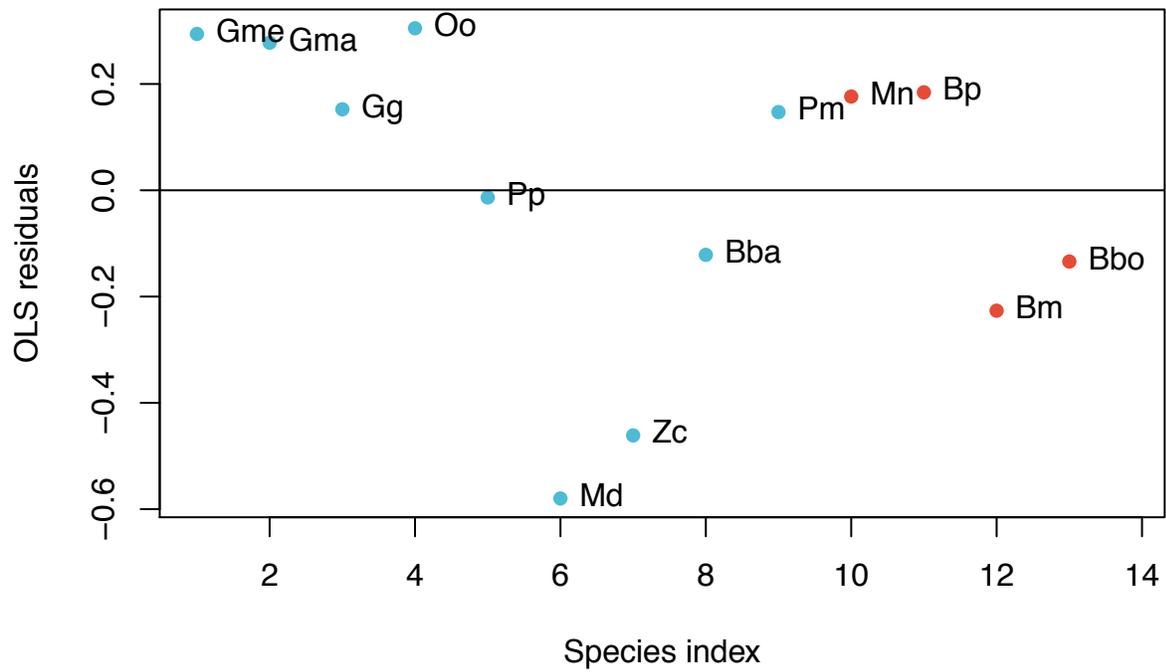
```
par(mfrow=c(1,2))
qqPlot(residuals(m.ols), id=FALSE)
plot(fitted(m.ols), residuals(m.ols))
abline(0,0)
```



7.3.3 Evaluate for phylogenetic correlation

7.3.3.1 Plot residuals ordered “by phylogeny” (i.e. in the order of tips of the phylogenetic tree)

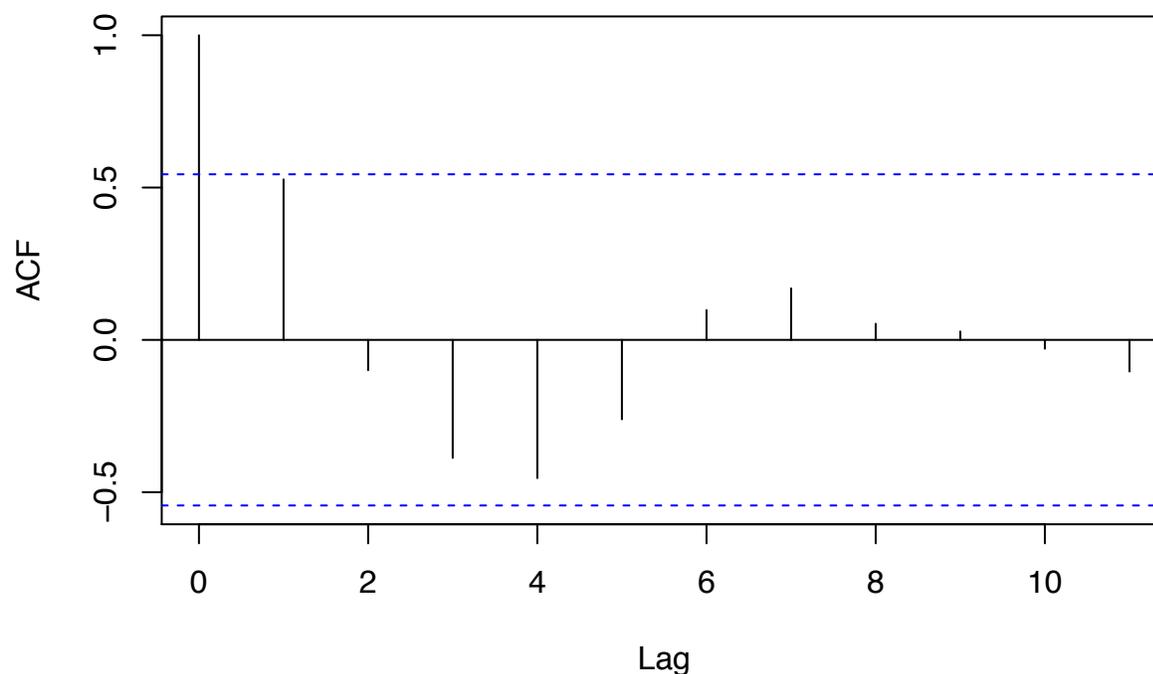
```
is_tip <- smytree$edge[,2]<=length(smytree$tip.label)
ordered_tips <- smytree$edge[is_tip,2]
oj <- residuals(m.ols)
tl <- smytree$tip.label[ordered_tips]
res <- oj[tl]
plot(oj[tl], pch=16, ylab="OLS residuals", xlab="Species index",
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata[tl,"fm"])],
     xlim=c(1,13.8))
abline(0,0)
text(oj[tl], labels=abbreviation[tl], pos=4)
```



7.3.3.2 Plot autocorrelation function of residuals ordered “by phylogeny”

```
acf(res, main="Series: residuals sorted by phylogeny")
```

Series: residuals sorted by phylogeny



7.4 Run a pGLS with feeding mode as a categorical predictor

7.4.1 Estimate Pagel's λ (amount of phylogenetic signal) for each trait separately

```
lambdax <- phylosig(smytree, smydata$x_mean, method = "lambda", test = T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
lambday <- phylosig(smytree, smydata$y_mean, method = "lambda", test = T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
cbind(lambdax, lambday)

##      lambdax      lambday
## lambda 1.014327  1.01842
## logL   -12.69024  -3.306273
## logL0  -17.41681  -14.50072
## P       0.002107892 2.226569e-06
```

7.4.2 Plot likelihood surface for Pagel's λ for model without feeding mode as a covariate

```
lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ x_mean, smydata,
  method = "REML", correlation = corPagel(value = lambda, phy = smytree,
  fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
```

```

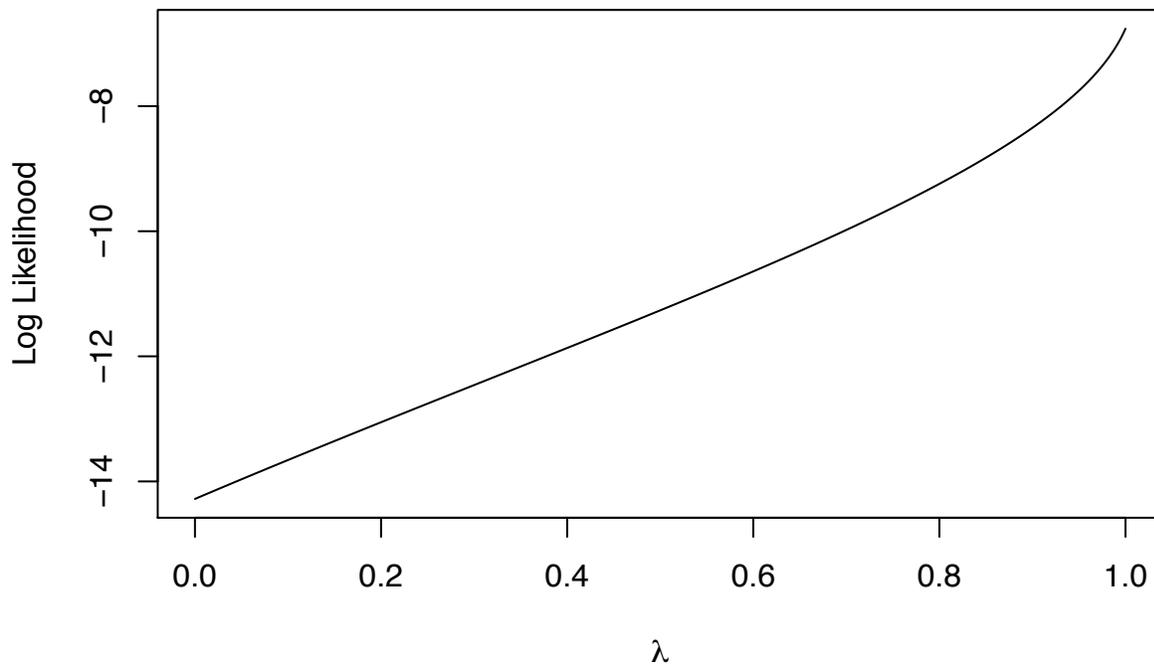
    expression(paste("Prey energy to body mass Likelihood Plot for ", lambda)),
    ylab = "Log Likelihood", xlab = expression(lambda))
m.pa.only <- gls(y_mean ~ x_mean, data = smydata, correlation =
                corPagel(value = 0, phy = smytree, fixed = FALSE), method = "REML")
m.pa.only$modelStruct[1]

## $corStruct
## Correlation structure of class corPagel representing
## lambda
## 2.129741

abline(v = m.pa.only$modelStruct[1], col = "red")

```

Prey energy to body mass Likelihood Plot for λ



7.4.3 Estimate Pagel's λ using REML

If λ is estimated to be greater than 1, fix it at 1, if smaller than 0, fix it at 0.

```

m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
                  corPagel(1, phy = smytree, fixed = FALSE), method = "REML")
summary(m.pgls.nlme)

## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC  logLik
## 8.682509 9.865857 1.658745
##

```

```

## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
## 1.018487
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.0532622 1.0882954 -0.9678092  0.3584
## fmSingle-prey  1.5132673 1.1033955  1.3714640  0.2034
## x_mean        0.6069504 0.2379275  2.5509884  0.0311
## fmSingle-prey:x_mean -0.6827300 0.2379275 -2.8694870  0.0185
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.986
## x_mean        -0.964  0.951
## fmSingle-prey:x_mean  0.964 -0.951 -1.000
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.3371829 -0.4379536  0.6567771  1.1109150  1.2362995
##
## Residual standard error: 0.3290555
## Degrees of freedom: 13 total; 9 residual

```

```
anova(m.pgls.nlme)
```

```

## Denom. DF: 9
##          numDF  F-value p-value
## (Intercept)    1      16  0.0033
## fm              1      17  0.0024
## x_mean          1 329195290 <.0001
## fm:x_mean       1       8  0.0185

```

```

lambda.est <- as.numeric(m.pgls.nlme$modelStruct[1])
if(lambda.est > 1){lambda.est <- 1} else if(lambda.est < 0){lambda.est <- 0}

```

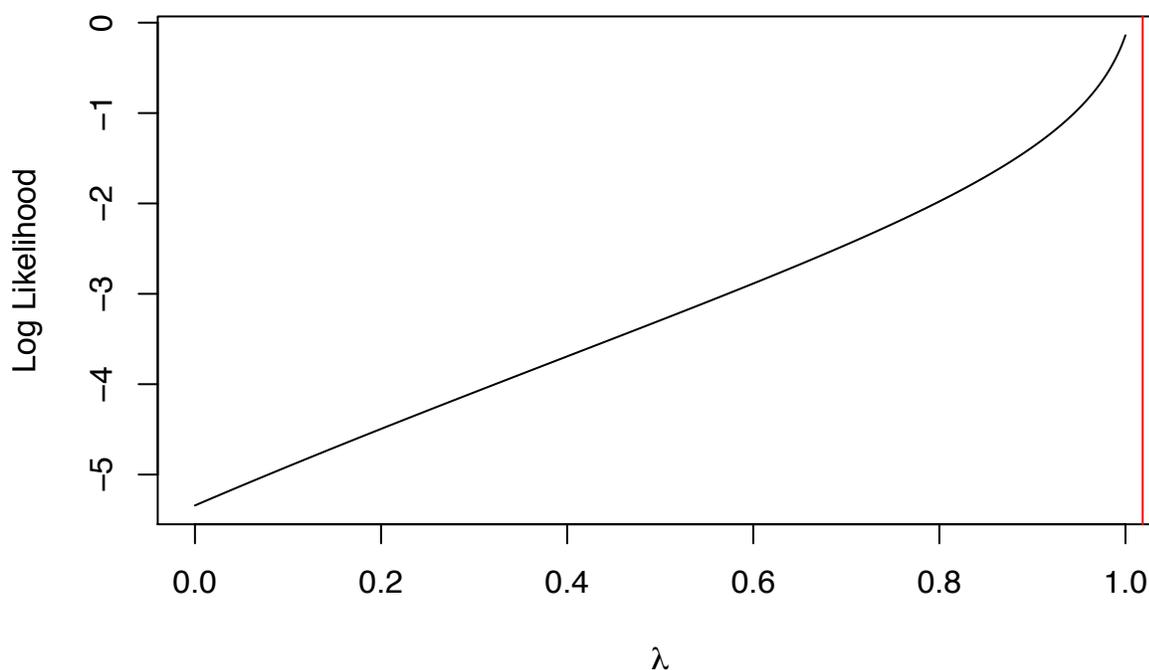
7.4.4 Plot likelihood surface for Pagel's λ - our estimate marked in red

```

lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ fm * x_mean, smydata,
                                                method = "REML", correlation =
                                                corPagel(value = lambda, phy = smytree, fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
     expression(paste("Energetic Efficiency to Body mass Likelihood Plot for ", lambda)),
     ylab = "Log Likelihood", xlab = expression(lambda)),
abline(v = m.pgls.nlme$modelStruct, col = "red")

```

Energetic Efficiency to Body mass Likelihood Plot for λ



7.4.5 Run pGLS and model reduction with a fixed Pagel's λ (using ML)

```
m.pgls.nlmf <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")  
summary(m.pgls.nlmf)
```

```
## Generalized least squares fit by maximum likelihood  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
## AIC BIC logLik  
## 3.5619 6.386646 3.21905  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
## 1  
##  
## Coefficients:  
## Value Std.Error t-value p-value  
## (Intercept) -1.0680820 1.0832731 -0.9859767 0.3499  
## fmSingle-prey 1.7245933 1.1590479 1.4879396 0.1709  
## x_mean 0.6107403 0.2370623 2.5762857 0.0299  
## fmSingle-prey:x_mean -0.7447819 0.2615778 -2.8472668 0.0192  
##  
## Correlation:
```

```

##          (Intr) fmSng- x_mean
## fmSingle-prey      -0.935
## x_mean             -0.966  0.903
## fmSingle-prey:x_mean 0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7362057 -0.5526648 0.7983987 1.1832279 1.4533689
##
## Residual standard error: 0.2681386
## Degrees of freedom: 13 total; 9 residual
anova(m.pgls.nlme)

## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)  1 16.705132 0.0027
## fm           1 18.404797 0.0020
## x_mean       1  0.000096 0.9924
## fm:x_mean    1  8.106928 0.0192

m.pgls.nlme.2 <- update(m.pgls.nlme, ~ . - fm:x_mean)
anova(m.pgls.nlme, m.pgls.nlme.2)

##      Model df      AIC      BIC  logLik  Test  L.Ratio
## m.pgls.nlme      1 5 3.561900 6.386646 3.219050
## m.pgls.nlme.2    2 4 9.911266 12.171064 -0.955633 1 vs 2 8.349367
##      p-value
## m.pgls.nlme
## m.pgls.nlme.2 0.0039

m.pgls.fm <- gls(y_mean ~ fm, data = smydata, correlation =
                corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")
summary(m.pgls.fm)

## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm
## Data: smydata
##      AIC      BIC      logLik
## 7.911339 9.606187 -0.9556694
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##      Value Std.Error  t-value p-value
## (Intercept) 1.627816 0.3493321 4.659795 0.0007
## fmSingle-prey -1.422404 0.4134742 -3.440128 0.0055
##
## Correlation:
##      (Intr)
## fmSingle-prey -0.845
##

```

```
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.3234807 -0.6768814  0.6336358  1.1560222  1.4386387
##
## Residual standard error: 0.3696794
## Degrees of freedom: 13 total; 11 residual
```

```
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC  logLik
##  3.5619  6.386646  3.21905
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -1.0680820  1.0832731 -0.9859767  0.3499
## fmSingle-prey   1.7245933  1.1590479  1.4879396  0.1709
## x_mean          0.6107403  0.2370623  2.5762857  0.0299
## fmSingle-prey:x_mean -0.7447819  0.2615778 -2.8472668  0.0192
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey   -0.935
## x_mean          -0.966  0.903
## fmSingle-prey:x_mean  0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7362057 -0.5526648  0.7983987  1.1832279  1.4533689
##
## Residual standard error: 0.2681386
## Degrees of freedom: 13 total; 9 residual
```

7.4.5.1 Compare to an intercept-only model

```
m.pgls.nlme.0 <- gls(y_mean ~ 1, smydata, correlation = corPagel(value = lambda.est,
  phy = smytree, fixed = TRUE), method = "ML")
anova(m.pgls.nlme, m.pgls.nlme.0)
```

```
##           Model df      AIC      BIC   logLik   Test  L.Ratio
## m.pgls.nlme     1  5  3.56190  6.386646  3.219050
## m.pgls.nlme.0   2  2 15.40623 16.536132 -5.703117 1 vs 2 17.84433
##
##           p-value
## m.pgls.nlme
## m.pgls.nlme.0  5e-04
```

```
m.pgls.p <- anova(m.pgls.nlme, m.pgls.nlme.0)$`p-value`[2]
```

7.4.6 Estimate final model using REML

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
                 corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "REML")  
summary(m.pgls.nlme)
```

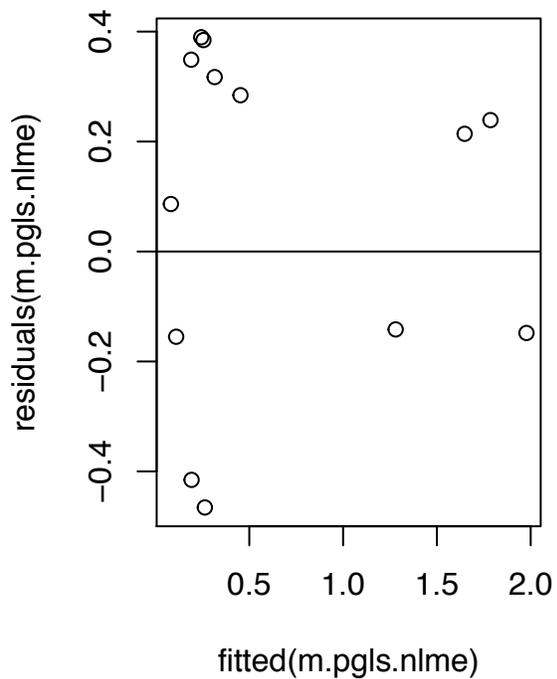
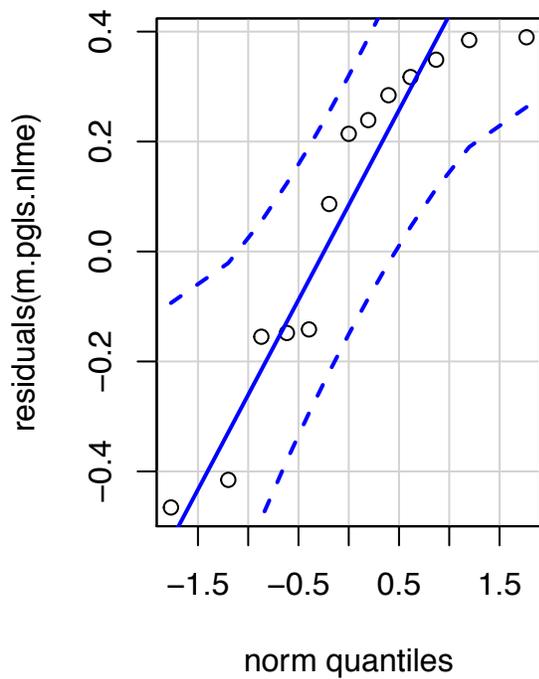
```
## Generalized least squares fit by REML  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
##      AIC      BIC      logLik  
## 10.27805 11.26418 -0.1390273  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
##      1  
##  
## Coefficients:  
##              Value Std.Error   t-value p-value  
## (Intercept)  -1.0680820 1.0832731 -0.9859767  0.3499  
## fmSingle-prey  1.7245933 1.1590479  1.4879396  0.1709  
## x_mean         0.6107403 0.2370623  2.5762857  0.0299  
## fmSingle-prey:x_mean -0.7447819 0.2615778 -2.8472668  0.0192  
##  
## Correlation:  
##              (Intr) fmSng- x_mean  
## fmSingle-prey  -0.935  
## x_mean         -0.966  0.903  
## fmSingle-prey:x_mean  0.875 -0.954 -0.906  
##  
## Standardized residuals:  
##      Min      Q1      Med      Q3      Max  
## -1.4446104 -0.4598449  0.6643078  0.9845051  1.2092760  
##  
## Residual standard error: 0.3222625  
## Degrees of freedom: 13 total; 9 residual
```

```
m.pgls.param <- as.data.frame(t(summary(m.pgls.nlme)$tTable[,1])) %>%  
  mutate(intercept.rorq = `(Intercept)`,  
         intercept.od = `(Intercept)` + `fmSingle-prey`,  
         slope.rorq = `x_mean`, slope.od = `x_mean` + `fmSingle-prey:x_mean`)  
m.pgls.param <- m.pgls.param[5:8]
```

7.4.6.1 Model diagnostics

7.4.6.1.1 QQ-plot and Residuals vs fitted plot

```
par(mfrow = c(1,2))  
qqPlot(residuals(m.pgls.nlme), id = FALSE)  
plot(fitted(m.pgls.nlme), residuals(m.pgls.nlme))  
abline(0,0)
```



7.5 Estimate confidence intervals by bootstrapping

7.5.1 Bootstrap and compute percentile confidence intervals

```
d_sub <- filter(d_full, MR.exponent==.68)
index <- d_sub %>% group_by(Spec) %>% summarize(ix=length(y))
index # number of prey categories for each species
```

```
## # A tibble: 13 x 2
##   Spec                ix
##   <fct>              <int>
## 1 Balaenoptera_bonaerensis    5
## 2 Balaenoptera_musculus      7
## 3 Balaenoptera_physalus      7
## 4 Berardius_bairdii         19
## 5 Globicephala_macrorhynchus 12
## 6 Globicephala_melas        12
## 7 Grampus_griseus           5
## 8 Megaptera_novaeangliae     8
## 9 Mesoplodon_densirostris    3
## 10 Orcinus_orca              12
## 11 Phocoena_phocoena         5
## 12 Physeter_macrocephalus    18
## 13 Ziphius_cavirostris       16
```

```

smydata.orig <- smydata
y_mean <- by(d_sub, d_sub$Spec, with, weighted.mean(y, Percent))

spec <- unique(d_sub$Spec)
spec <- spec[match(spec, smydata$species)]

runpGls <- function(smydata, smytree){
  out <- tryCatch(
    {
      model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
        corPagel(lambda.est, phy = smytree, fixed = FALSE),
        method = "REML")
      as.numeric(model.pgls$modelStruct[1])
    },
    error=function(cond) {
      return(NA)
    }
  )
}

a.ols <- matrix(nrow=10000, ncol=4)
a.pgls <- matrix(nrow=10000, ncol=4)
b <- matrix(nrow=10000, ncol=length(spec))
boot.lambdas <- rep(NA, 10000)
for(i in 1:10000){
  for(sp in 1:length(spec)){
    ix <- sample(1:index$ix[index$Spec==spec[sp]], replace = T)
    y_mean[sp] <- sum(d_sub[d_sub$Spec==spec[sp], "y"][ix] *
      d_sub[d_sub$Spec==spec[sp], "Percent"][ix]) /
      sum(d_sub[d_sub$Spec==spec[sp], "Percent"][ix])
  }
  smydata$y_mean <- y_mean

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
  myout <- runpGls(smydata, smytree)
  boot.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(lambda.est, phy = smytree, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(l.est, phy = smytree, fixed = TRUE), method = "REML")
  }

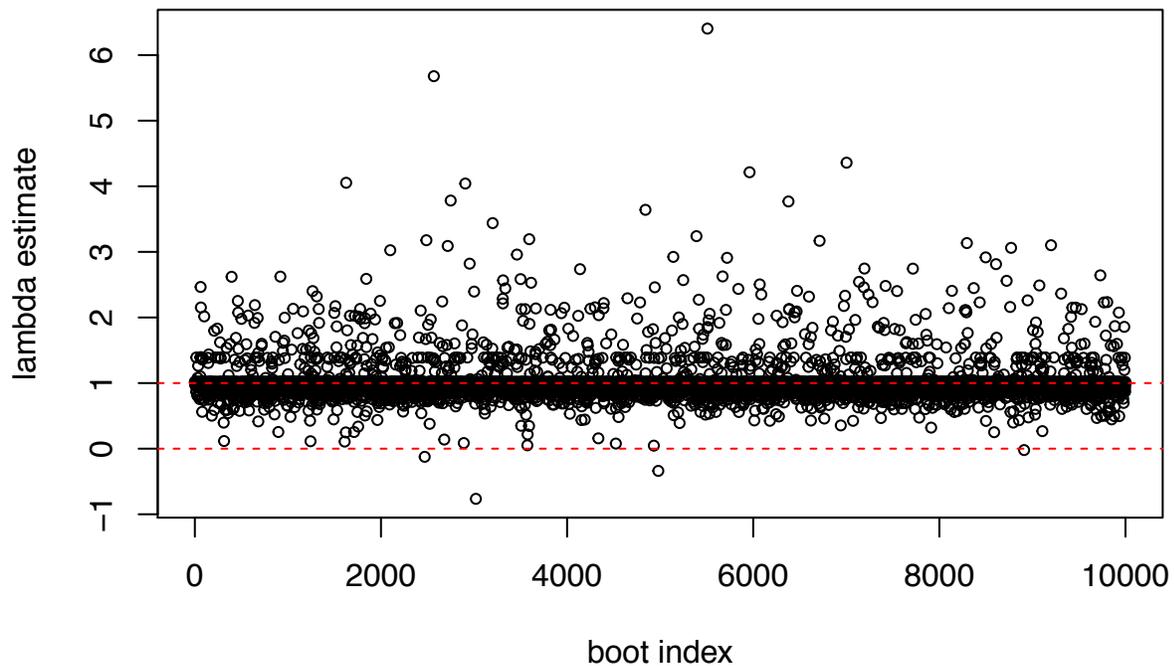
  a.ols[i,] <- c(coef(model.ols)[1], coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3], coef(model.ols)[3]+coef(model.ols)[4])
  a.pgls[i,] <- c(coef(model.pgls)[1], coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3], coef(model.pgls)[3]+coef(model.pgls)[4])
  b[i,] <- predict(model.ols)
}

```

```
# number of pGLS models, where lambda could not be estimated ==> used original value:
sum(is.na(boot.lambdas))
```

```
## [1] 209
```

```
plot(boot.lambdas, cex=.7, xlab="boot index", ylab="lambda estimate")
abline(h=0,lty="dashed",col="red")
abline(h=1,lty="dashed",col="red")
```



```
preds <- apply(b, 2, quantile, c(0.025, 0.975))
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_68_bootstrap_b.rds")
saveRDS(b,fil)
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_68_bootstrap_preds.rds")
saveRDS(preds,fil)

df.boot.ols <- data.frame(cbind(t(m.ols.param),t(t(apply(a.ols, 2, mean))),
                             t(apply(a.ols, 2, quantile, c(0.025, 0.975))))))
names(df.boot.ols) <- c("obs","bootest","lowerCI","upperCI")
df.boot.pgls <- data.frame(cbind(t(m.pgls.param),t(t(apply(a.pgls, 2, mean))),
                             t(apply(a.pgls, 2, quantile, c(0.025, 0.975))))))
names(df.boot.pgls) <- c("obs","bootest","lowerCI","upperCI")

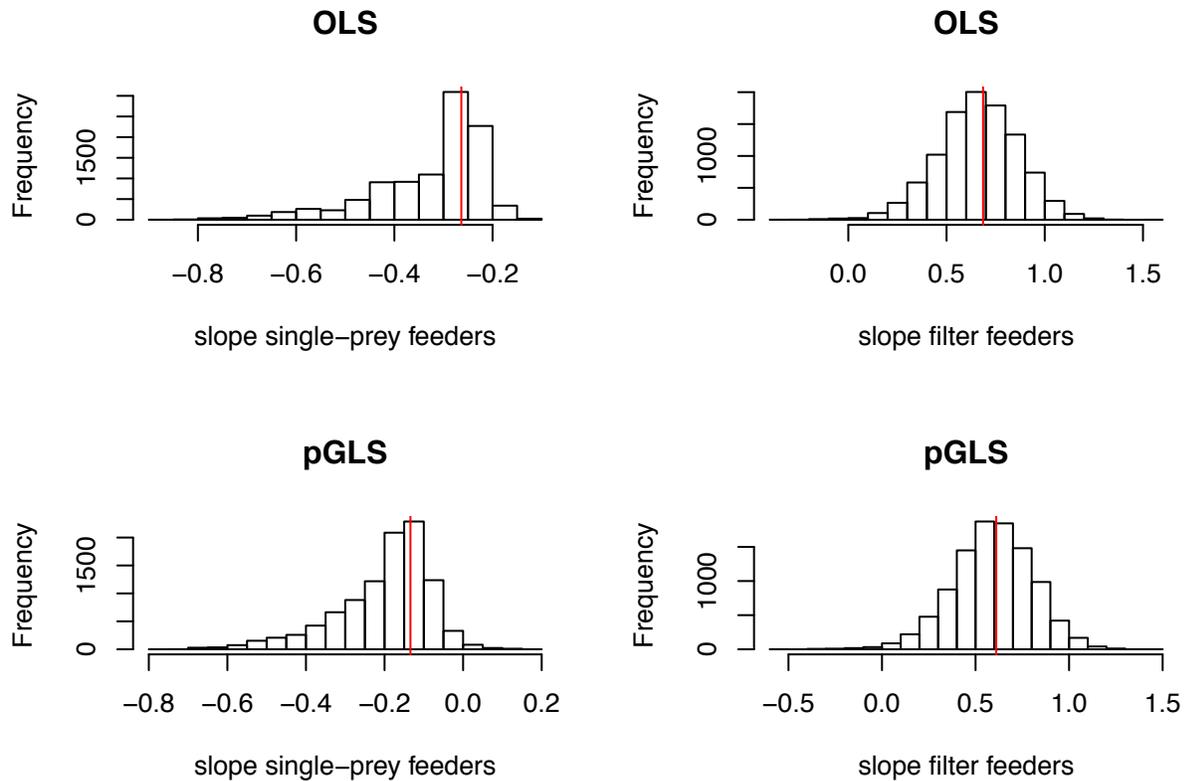
par(mfrow=c(2,2))
hist(a.ols[,4], xlab="slope single-prey feeders", main="OLS")
```

```

abline(v=m.ols.param[4], col="red")
hist(a.ols[,3], xlab="slope filter feeders", main="OLS")
abline(v=m.ols.param[3], col="red")

hist(a.pglis[,4], xlab="slope single-prey feeders", main="pGLS")
abline(v=m.pglis.param[4], col="red")
hist(a.pglis[,3], xlab="slope filter feeders", main="pGLS")
abline(v=m.pglis.param[3], col="red")

```



7.5.2 Compute BCa (bias-corrected and accelerated) confidence intervals

```

smydata <- smydata.orig

# compute bias-correction factor from the proportion of bootstrap estimates
# that are less than the observed estimate

bootBC <- function(bootEst, Est){
  B <- ncol(bootEst)*nrow(bootEst) # number of bootstrap samples
  propLess <- sum(bootEst < Est)/B # proportion of replicates less than observed stat
  z0 <- qnorm(propLess) # bias correction
  return(z0)
}

z0.ols <- numeric()
for (i in 1:ncol(a.ols)){

```

```

z0.ols[i] <- bootBC(t(t(a.ols[,i])),as.numeric(m.ols.param[i]))
}

z0.pgls <- numeric()
for (i in 1:ncol(a.pgls)){
z0.pgls[i] <- bootBC(t(t(a.pgls[,i])),as.numeric(m.pgls.param[i]))
}

# compute acceleration factor, which is related to the skewness of bootstrap estimates.
# Use jackknife replicates to estimate.

jStat.ols <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jStat.pgls <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jack.lambdas <- rep(NA,nrow(smydata))
for (i in 1:nrow(smydata)) {
  d_sub <- subset(d_full, Spec==smydata$species[i] & MR.exponent==.68)
  y_mean.j <- numeric()
  for(j in 1:nrow(d_sub)){
    d_sub.j <- d_sub[-j,]
    y_mean.j[j] <- sum(d_sub.j$y*d_sub.j$Percent)/sum(d_sub.j$Percent)
  }
  smydata.j <- smydata
  smydata.j$y_mean[i] <- mean(y_mean.j)
  pruned.tree <- drop.tip(smytree,smytree$tip.label[-match(smydata.j$species,
    smytree$tip.label)])

  smytree.j <- pruned.tree
  smydata.j <- smydata.j[match(smytree.j$tip.label,rownames(smydata.j)),]

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata.j, method = "REML")

  myout <- runpGls(smydata.j,smytree.j)
  jack.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(lambda.est, phy = smytree.j, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(l.est, phy = smytree.j, fixed = TRUE), method = "REML")
  }

  jStat.ols[,i] <- as.numeric(c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4]))
  jStat.pgls[,i] <- as.numeric(c(coef(model.pgls)[1],
    coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],
    coef(model.pgls)[3]+coef(model.pgls)[4]))
}
jackEst.ols <- t(t(apply(jStat.ols, 1, mean))) # jackknife estimate

```

```

jackEst.pgls <- t(t(apply(jStat.pgls, 1, mean))) # jackknife estimate
jack.lambdas # lambdas of the jackknifed models

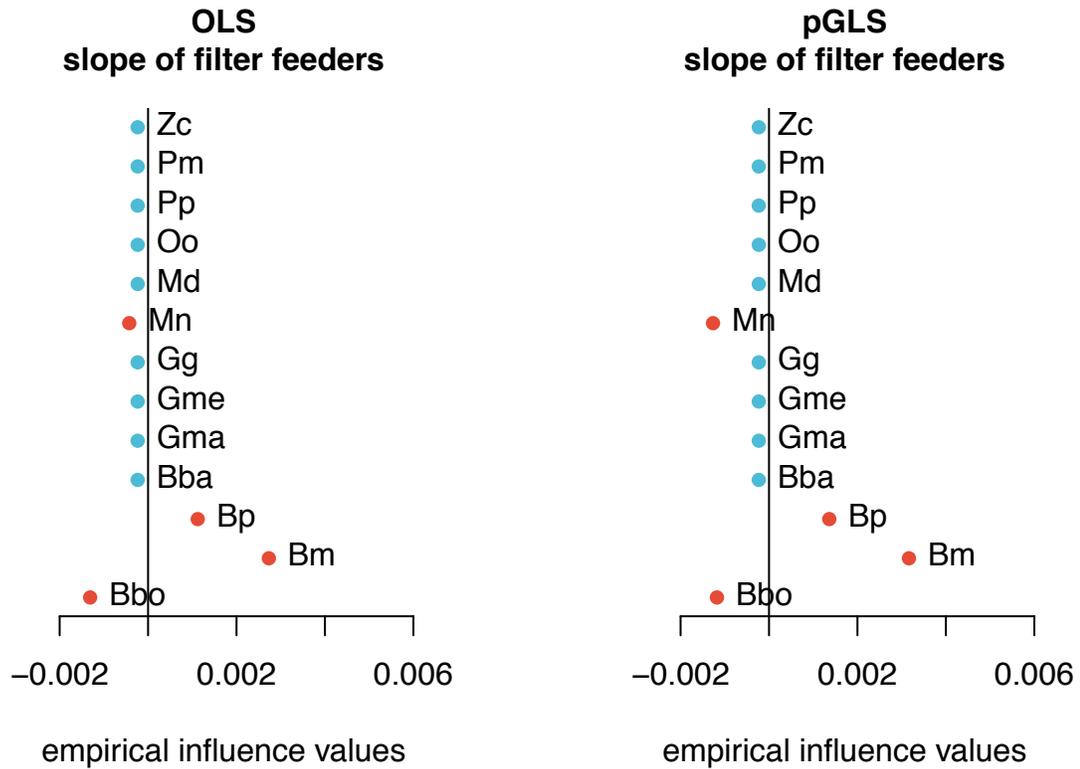
## [1] 1.018487 1.018487 1.018487 1.018487 1.018487 1.018487 1.018487
## [8] 1.018487 1.018487 1.018487 1.018487 1.018487 1.018487

num.ols <- numeric(); den.ols <- numeric(); ahat.ols <- numeric()
num.pgls <- numeric(); den.pgls <- numeric(); ahat.pgls <- numeric()
for (i in 1:nrow(jStat.ols)){
  num.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^3 )
  den.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^2 )
  ahat.ols[i] <- num.ols[i]/(6*den.ols[i]^(3/2)) # ahat based on jackknife
  num.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^3 )
  den.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^2 )
  ahat.pgls[i] <- num.pgls[i]/(6*den.pgls[i]^(3/2)) # ahat based on jackknife
}

# influential species:
par(mfrow=c(1,2))
plot(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("OLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("pGLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```

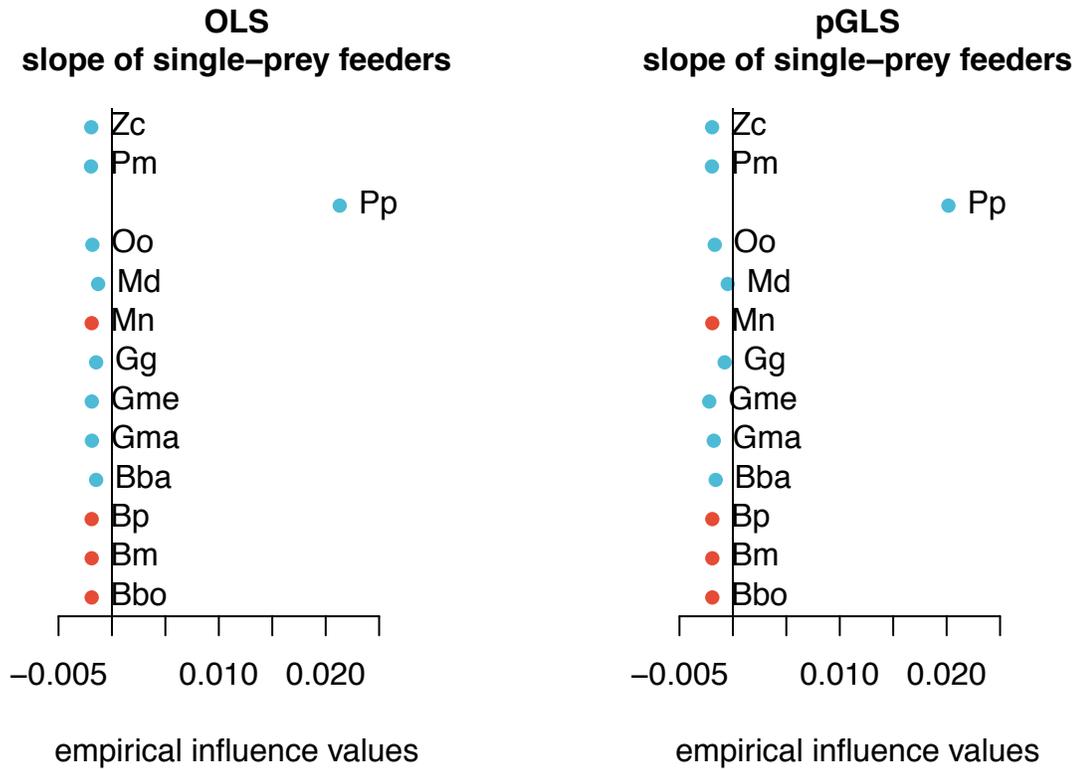


```

par(mfrow=c(1,2))
plot(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("OLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("pGLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```



```
# adjust quantiles for 100*(1-alpha)% bootstrap BCa interval
```

```
alpha <- 0.05
zL.ols <- z0.ols + qnorm(alpha/2)
alpha1.ols <- pnorm(z0.ols + zL.ols / (1-ahat.ols*zL.ols))
zU.ols <- z0.ols + qnorm(1-alpha/2)
alpha2.ols <- pnorm(z0.ols + zU.ols / (1-ahat.ols*zU.ols))

zL.pgls <- z0.pgls + qnorm(alpha/2)
alpha1.pgls <- pnorm(z0.pgls + zL.pgls / (1-ahat.pgls*zL.pgls))
zU.pgls <- z0.pgls + qnorm(1-alpha/2)
alpha2.pgls <- pnorm(z0.pgls + zU.pgls / (1-ahat.pgls*zU.pgls))
```

```
cbind((alpha1.ols*100), (alpha2.ols*100)) # new quantiles OLS:
```

```
##           [,1]      [,2]
## [1,]  0.497674069  93.53320
## [2,]  0.007961714  83.75655
## [3,]  6.957140477  99.58687
## [4,] 16.788999767  99.99338
```

```
cbind((alpha1.pgls*100), (alpha2.pgls*100)) # new quantiles pGLS:
```

```
##           [,1]      [,2]
## [1,]  0.713765473  94.49841
## [2,]  0.001266081  77.60691
## [3,]  5.846356114  99.39407
```

```
## [4,] 22.917289319 99.99893
CI.ols <- matrix(nrow = ncol(a.ols), ncol=2)
for (i in 1:ncol(a.ols)){
  CI.ols[i,] <- quantile(a.ols[,i], c(alpha1.ols[i], alpha2.ols[i])) # BCa interval
}
df.boot.ols$lowerCIbca <- CI.ols[,1]
df.boot.ols$upperCIbca <- CI.ols[,2]

CI.pgls <- matrix(nrow = ncol(a.pgls), ncol=2)
for (i in 1:ncol(a.pgls)){
  CI.pgls[i,] <- quantile(a.pgls[,i], c(alpha1.pgls[i], alpha2.pgls[i])) # BCa interval
}
df.boot.pgls$lowerCIbca <- CI.pgls[,1]
df.boot.pgls$upperCIbca <- CI.pgls[,2]
```

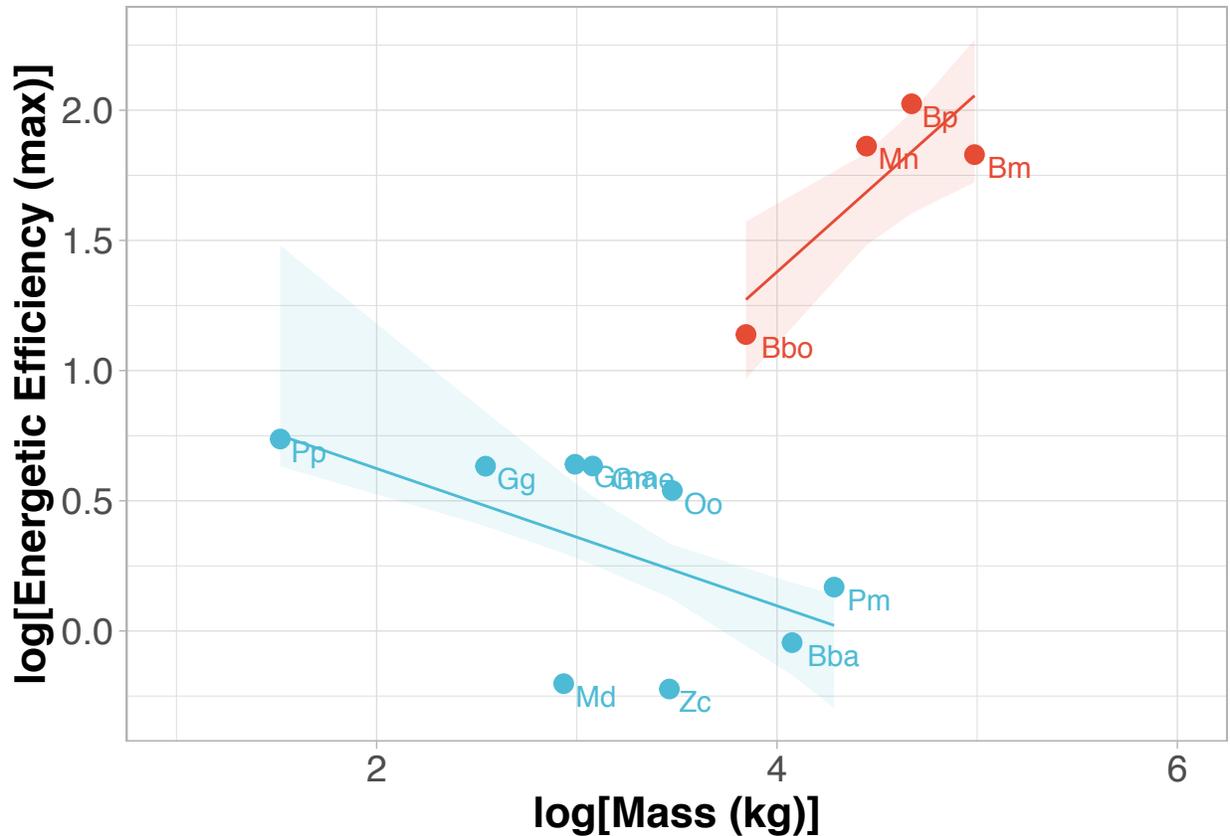
7.5.3 Plot OLS model

```
smydata <- smydata.orig

ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit,
                                       y_min = preds[1,], y_max = preds[2,]))

fig_ols <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
           linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
           linetype = 1) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Rorqual"), fill = "#E64B35FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Odontocete"), fill = "#4DBBD5FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
           aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
           aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_ols
```



7.5.3.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCI"],
  lowerCI.od = df.boot.ols["slope.od","lowerCI"],
  upperCI.od = df.boot.ols["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.ols["slope.od","lowerCIbca"],
  upperCI.od = df.boot.ols["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.ols[,3],
  odontocete_slope=a.ols[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF"), color = NA, alpha = 0.3) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF"), color = NA, alpha = 0.3) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

```

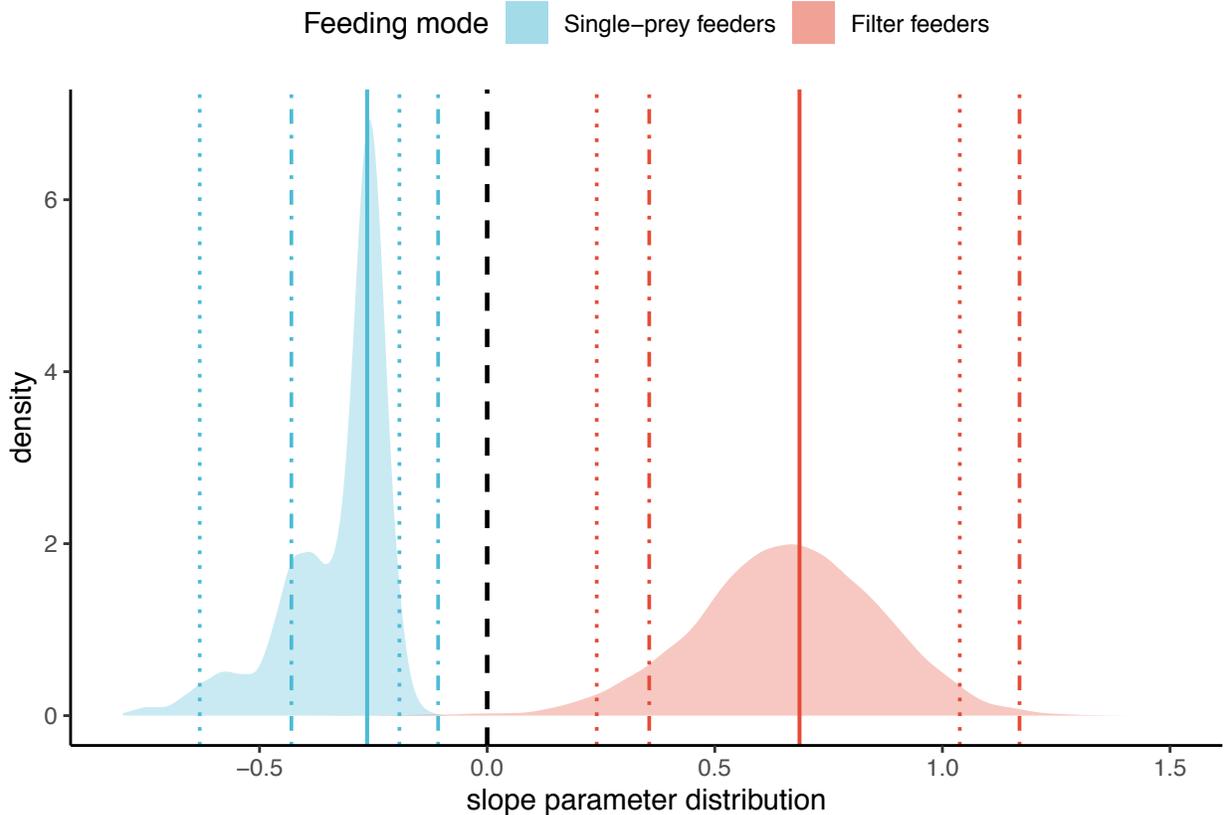
```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 8 rows containing non-finite values (stat_density).

Warning: Removed 1 rows containing non-finite values (stat_density).



```
rn <- rownames(df.boot.ols)
rownames(df.boot.ols) <- c("intercept filter","intercept single-prey",
                          "slope filter","slope single-prey")
knitr::kable(df.boot.ols,
             caption = "OLS 95%\% Bootstrap Pctl and BCa CI",
             format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
kable_styling(latex_options = "hold_position")
```

Table 15: OLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-1.3654	-1.2640	-2.9795	0.5871	-3.5212	0.1441
intercept single-prey	1.1510	1.3683	0.9384	2.4369	0.6359	1.7456
slope filter	0.6861	0.6599	0.2407	1.0382	0.3560	1.1694
slope single-prey	-0.2636	-0.3263	-0.6314	-0.1928	-0.4300	-0.1077

```
rownames(df.boot.ols) <- rn
```

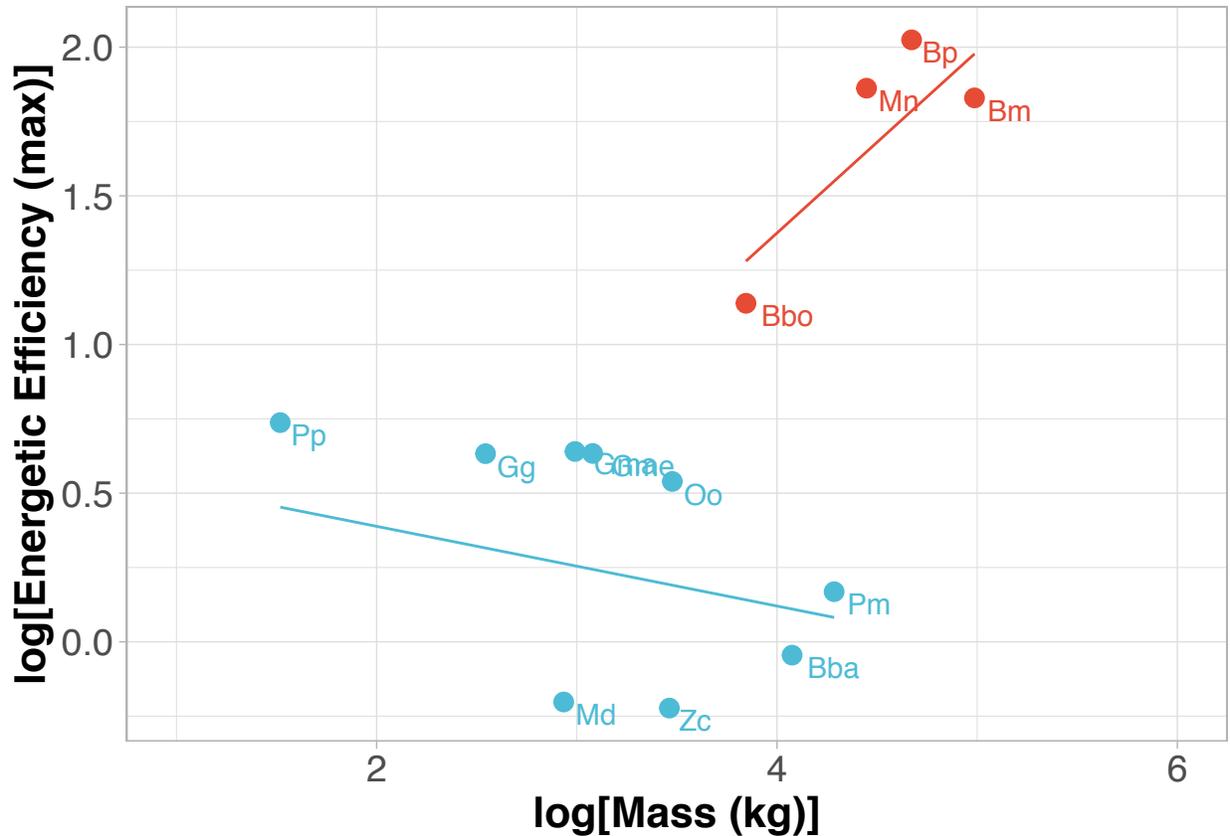
7.5.4 Plot pGLS model

```
pgls.fit <- predict(m.pgls.nlme)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))

fig_pgls <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
            linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
            linetype = 1) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +

  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_pgls
```



7.5.4.1 Plot kernel density distributions of slopes

```

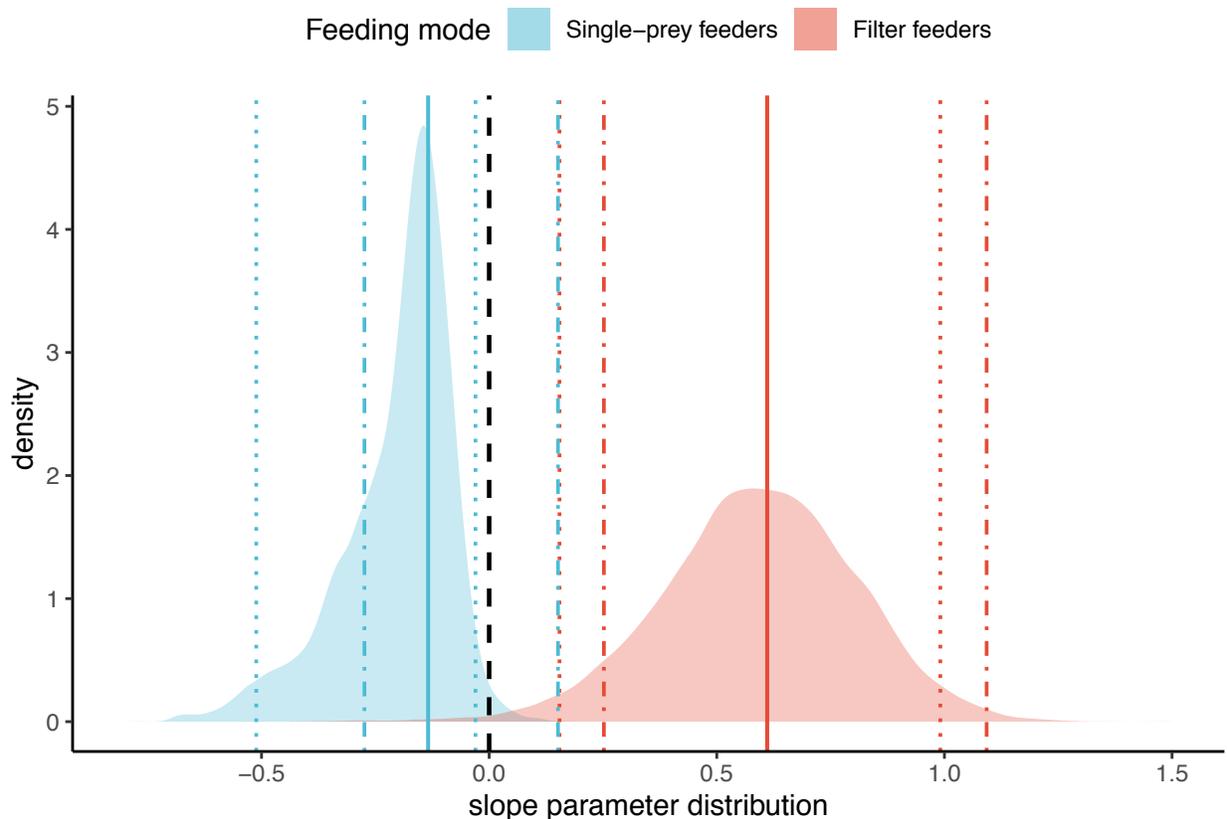
model_param <- data.frame(slope.rorq = df.boot.pglsl["slope.rorq","obs"],
  slope.od = df.boot.pglsl["slope.od","obs"],
  lowerCI.rorq = df.boot.pglsl["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.pglsl["slope.rorq","upperCI"],
  lowerCI.od = df.boot.pglsl["slope.od","lowerCI"],
  upperCI.od = df.boot.pglsl["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.pglsl["slope.rorq","obs"],
  slope.od = df.boot.pglsl["slope.od","obs"],
  lowerCI.rorq = df.boot.pglsl["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.pglsl["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.pglsl["slope.od","lowerCIbca"],
  upperCI.od = df.boot.pglsl["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.pglsl[,3],
  odontocete_slope=a.pglsl[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF", color = NA, alpha = 0.3)) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF", color = NA, alpha = 0.3)) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

```

```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
  geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```



```

rn <- rownames(df.boot.pgls)
rownames(df.boot.pgls) <- c("intercept filter","intercept single-prey",
  "slope filter","slope single-prey")
knitr::kable(df.boot.pgls,

```

```
caption = "pGLS 95% Bootstrap Pctl and BCa CI",
format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
kable_styling(latex_options = "hold_position")
```

Table 16: pGLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-1.0681	-0.9959	-2.7980	0.9239	-3.2324	0.5368
intercept single-prey	0.6565	0.9070	0.3163	2.0848	-0.2993	1.1853
slope filter	0.6107	0.5921	0.1544	0.9910	0.2521	1.0927
slope single-prey	-0.1340	-0.2013	-0.5115	-0.0300	-0.2740	0.1512

```
rownames(df.boot.pgls) <- rn
```

7.6 Extract summary statistics

```
specify_decimal <- function(x, k) trimws(format(round(x, k), nsmall = k))

res.df.ols <- m.ols$dim$N - m.ols$dim$p

res.df.pgls <- m.pgls.nlme$dim$N - m.pgls.nlme$dim$p

intercepts.od.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCIbca"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCIbca"], 4),
  ")"))

intercepts.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCIbca"],
  4), ")"),
  paste0(specify_decimal(df.boot.ols["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.rorq", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.rorq", "lowerCIbca"], 4),
```

```

" - ", specify_decimal(df.boot.ols["intercept.rorq", "upperCIbca"], 4),
")"))

slopes.od.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.od", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.pgl["slope.od", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.pgl["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.od", "lowerCIbca"], 4), " - ",
  specify_decimal(df.boot.pgl["slope.od", "upperCIbca"], 4), ")"),
paste0(specify_decimal(df.boot.ols["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.od", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.ols["slope.od", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.ols["slope.od", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.od", "lowerCIbca"], 4), " - ",
  specify_decimal(df.boot.ols["slope.od", "upperCIbca"], 4), ")"))

slopes.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgl["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.rorq", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.pgl["slope.rorq", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.pgl["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.pgl["slope.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgl["slope.rorq", "upperCIbca"],
  4), ")"),
paste0(specify_decimal(df.boot.ols["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.rorq", "lowerCI"], 4), " - ",
  specify_decimal(df.boot.ols["slope.rorq", "upperCI"], 4), ")"),
paste0(specify_decimal(df.boot.ols["slope.rorq", "obs"], 4), " (",
  specify_decimal(df.boot.ols["slope.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.ols["slope.rorq", "upperCIbca"],
  4), ")"))

a.od.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.pgl["intercept.od", "lowerCI"]), 4),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.od", "upperCI"]), 4),
  ")"),
paste0(specify_decimal(10^(df.boot.pgl["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.pgl["intercept.od", "lowerCIbca"]), 4),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.od", "upperCIbca"]), 4),
  ")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.ols["intercept.od", "lowerCI"]), 4), " - ",
  specify_decimal(10^(df.boot.ols["intercept.od", "upperCI"]), 4), ")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od", "obs"]), 4), " (",
  specify_decimal(10^(df.boot.ols["intercept.od", "lowerCIbca"]), 4),
  " - ", specify_decimal(10^(df.boot.ols["intercept.od", "upperCIbca"]),
  4), ")"))

a.rorq.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq", "obs"]), 4),
  " (", specify_decimal(10^(df.boot.pgl["intercept.rorq", "lowerCI"]), 5),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.rorq", "upperCI"]), 4), ")"),
paste0(specify_decimal(10^(df.boot.pgl["intercept.rorq", "obs"]), 4),
  " (", specify_decimal(10^(df.boot.pgl["intercept.rorq", "lowerCIbca"]), 5),
  " - ", specify_decimal(10^(df.boot.pgl["intercept.rorq", "upperCIbca"]), 4),
  ")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.rorq", "obs"]), 4),

```

Table 17: Model summary statistics

	<i>Filter feeders</i>		<i>Single-prey feeders</i>		RSE	tot.df	res.df
	slope*	intercept	slope	intercept			
pGLS	0.6107 (0.1544 - 0.9910)	-1.0681 (-2.7980 - 0.9239)	-0.1340 (-0.5115 - -0.0300)	0.6565 (0.3163 - 2.0848)	0.3223		
	0.6107 (0.2521 - 1.0927)	-1.0681 (-3.2324 - 0.5368)	-0.1340 (-0.2740 - 0.1512)	0.6565 (-0.2993 - 1.1853)	0.3223		
OLS	0.6861 (0.2407 - 1.0382)	-1.3654 (-2.9795 - 0.5871)	-0.2636 (-0.6314 - -0.1928)	1.1510 (0.9384 - 2.4369)	0.3333	13	9
	0.6861 (0.3560 - 1.1694)	-1.3654 (-3.5212 - 0.1441)	-0.2636 (-0.4300 - -0.1077)	1.1510 (0.6359 - 1.7456)	0.3333		

Note:

* Throughout the table, values in brackets represent 95% confidence intervals: percentile in shaded rows, BCa in non-shaded rows.

```

" (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCI"]),5),
" - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCI"]),4),")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCIbca"]),5),
" - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCIbca"]),4),
")"))

RSE <- rbind(specify_decimal(t(t(rep(as.numeric(m.pgls.nlm$sigma),2))),4),
             specify_decimal(t(t(rep(as.numeric(m.ols$sigma),2))),4))
df <- cbind(t(t(c(rep(m.pgls.nlm$dims$N,2), rep(m.ols$dims$N,2)))),
            t(t(c(rep(res.df.pgls,2), rep(res.df.ols,2)))))
models <- rbind(t(t(rep("pGLS",2))),t(t(rep("OLS",2))))

outputs <- cbind(models, slopes.rorq.ci, intercepts.rorq.ci, slopes.od.ci,
                 intercepts.od.ci, RSE, df)
df.outputs <- data.frame(outputs, check.rows = TRUE, check.names = TRUE)
names(df.outputs) <- c("","slope","intercept","slope","intercept","RSE", "tot.df","res.df")
names(df.outputs)[2] <- paste0(names(df.outputs)[2],
                               footnote_marker_symbol(1))

knitr::kable(df.outputs,
             caption = "Model summary statistics",
             format = "latex", booktabs = TRUE, escape = FALSE) %>%
kable_styling(latex_options = "scale_down") %>%
row_spec(0, bold = T) %>%
row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
column_spec(c(1,(ncol(df.outputs)-1):ncol(df.outputs))-1,
            background = "white") %>%
column_spec(1, bold = T) %>%
collapse_rows(columns = c(1,(ncol(df.outputs)-1):ncol(df.outputs))) %>%
add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2,
                  " " = 3), bold = T, italic = T) %>%
footnote(general = "", general_title = "Note:",
         symbol = paste0("Throughout the table, values in brackets",
                        " represent 95% confidence intervals: ",
                        "percentile in shaded rows, BCa in non-shaded rows."),
         symbol_title = "", title_format = "italic",
         footnote_as_chunk = T)

alloout <- cbind(models, a.rorq.ci, slopes.rorq.ci, a.od.ci, slopes.od.ci)
df.allo <- data.frame(alloout, check.rows = TRUE, check.names = TRUE)
names(df.allo) <- c("","a","b","a","b")
names(df.allo)[2] <- paste0(names(df.allo)[2], footnote_marker_symbol(1))
knitr::kable(df.allo,

```

Table 18: Transformed to allometric equations

	<i>Filter feeders</i>		<i>Single-prey feeders</i>	
	a*	b	a	b
pGLS	0.0855 (0.00159 - 8.3920)	0.6107 (0.1544 - 0.9910)	4.5343 (2.0713 - 121.5747)	-0.1340 (-0.5115 - -0.0300)
	0.0855 (0.00059 - 3.4418)	0.6107 (0.2521 - 1.0927)	4.5343 (0.5021 - 15.3209)	-0.1340 (-0.2740 - 0.1512)
OLS	0.0431 (0.00105 - 3.8645)	0.6861 (0.2407 - 1.0382)	14.1592 (8.6784 - 273.4721)	-0.2636 (-0.6314 - -0.1928)
	0.0431 (3e-04 - 1.3935)	0.6861 (0.3560 - 1.1694)	14.1592 (4.3243 - 55.6615)	-0.2636 (-0.4300 - -0.1077)

* Throughout the table, values in brackets represent 95% confidence intervals.: percentile in shaded rows, BCa in non-shaded rows.

```
caption = "Transformed to allometric equations",
format = "latex", booktabs = TRUE, escape = FALSE) %>%
kable_styling(latex_options = "scale_down") %>%
row_spec(0, bold = T) %>%
row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
column_spec(1, bold = T) %>%
collapse_rows(columns = 1) %>%
add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2),
bold = T, italic = T) %>%
footnote(symbol = paste0("Throughout the table, values in brackets",
" represent 95% confidence intervals.: ",
"percentile in shaded rows, BCa in non-shaded rows."),
symbol_title = "", threeparttable = TRUE, footnote_as_chunk = T)
```

7.7 Plot best models (OLS - dashed, PGLS - solid)

```
pgls.fit <- predict(m.pgls.nlme)
ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))

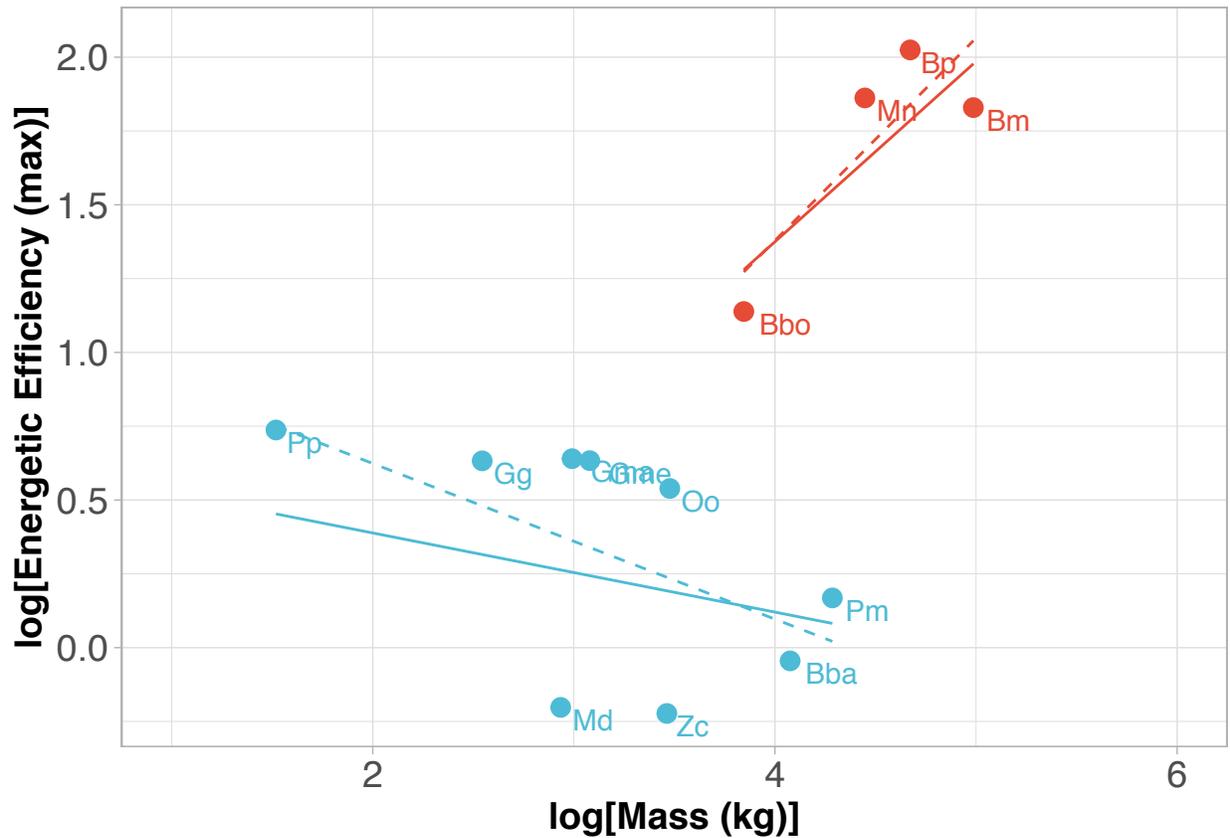
fig_4.68 <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
linetype = 1) +
  geom_line(data = dplyr::filter(predframe2, Group == "Rorqual"), color = "#E64B35FF",
linetype = 2) +
  geom_line(data = dplyr::filter(predframe2, Group == "Odontocete"), color = "#4DBBD5FF",
linetype = 2) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  xlim(1,6) +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 14,
face = "bold")) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
```

```

geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
          aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
          aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

```

fig_4.68



7.7.1 Construct output table

```

df.out <- smydata[,c("species", "fm", "x_mean", "y_mean")]
df.out$fitted_ols <- fitted(m.ols)
df.out$fitted_pgls <- fitted(m.pgls.nlme)
rownames(df.out) <- NULL
kable(df.out,
      caption = "Model outputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")

```

7.8 Quick clean up

```

m.68.pgls.nlme <- m.pgls.nlme
df.68.outputs <- df.outputs
m.68.ols <- m.ols
to.keep <- c(to.keep, "m.68.pgls.nlme", "df.68.outputs", "m.68.ols")
rm(list=setdiff(ls(), to.keep))

```

Table 19: Model outputs

species	fm	x_mean	y_mean	fitted_ols	fitted_pgl
Balaenoptera_bonaerensis	Filter	3.8451	1.1386	1.2726	1.2803
Balaenoptera_musculus	Filter	4.9868	1.8293	2.0559	1.9775
Balaenoptera_physalus	Filter	4.6725	2.0246	1.8403	1.7856
Berardius_bairdii	Single-prey	4.0755	-0.0448	0.0767	0.1102
Globicephala_macrorhynchus	Single-prey	2.9912	0.6401	0.3625	0.2556
Globicephala_melas	Single-prey	3.0792	0.6335	0.3393	0.2438
Grampus_griseus	Single-prey	2.5441	0.6328	0.4804	0.3155
Megaptera_novaeangliae	Filter	4.4472	1.8621	1.6857	1.6480
Mesoplodon_densirostris	Single-prey	2.9345	-0.2024	0.3775	0.2632
Orcinus_orca	Single-prey	3.4771	0.5393	0.2344	0.1904
Phocoena_phocoena	Single-prey	1.5185	0.7372	0.7507	0.4530
Physeter_macrocephalus	Single-prey	4.2856	0.1685	0.0213	0.0821
Ziphius_cavirostris	Single-prey	3.4624	-0.2229	0.2383	0.1924

8 Run model for MR = .75

8.1 Prepare data

8.1.1 Get rid of rows with NAs - subset the data

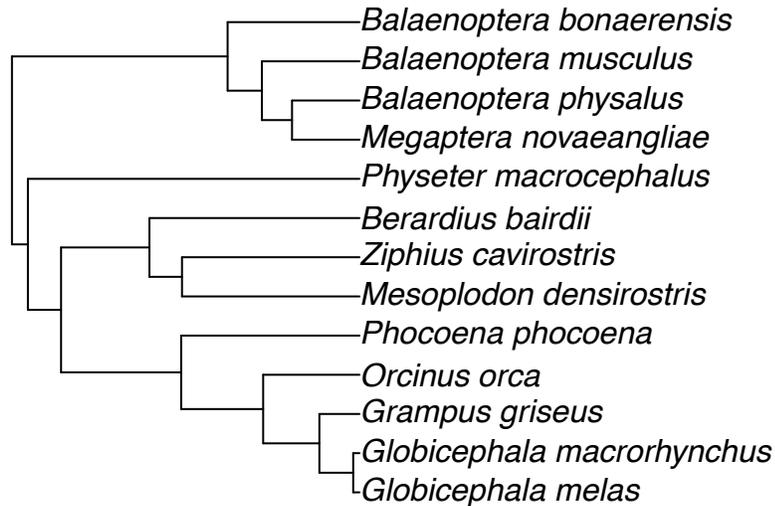
```
smydata <- df.spec
smydata$y_mean <- smydata$wgtMean.75
smydata <- smydata[!is.na(smydata$y_mean),]
smydata <- smydata[!is.na(smydata$x_mean),]
smydata$fm <- factor(smydata$fm)
smydata$Group <- smydata$Group
colnames(smydata)[1] <- "species"
```

8.1.2 Adjust tree - drop species for which data are missing

```
smytree <- drop.tip(mytree,mytree$tip.label[!match(smydata$species, mytree$tip.label)])
plot(smytree)
```

Table 20: Model inputs

species	gr	x_mean	fm	Group	abbreviation	wgtMean.45	wgtMean.61	wgtMean.68	wgtMean.75	y_mean
<i>Balaenoptera bonaerensis</i>	1	3.8451	Filter	Rorqual	Bbo	1.3289	1.2293	1.1386	1.0061	1.0061
<i>Balaenoptera musculus</i>	1	4.9868	Filter	Rorqual	Bm	2.0165	1.9307	1.8293	1.6606	1.6606
<i>Balaenoptera physalus</i>	1	4.6725	Filter	Rorqual	Bp	2.2749	2.1286	2.0246	1.8348	1.8348
<i>Berardius bairdii</i>	5	4.0755	Single-prey	Odontocete	Bba	0.7242	-0.3180	-0.0448	-0.3180	-0.3180
<i>Globicephala macrorhynchus</i>	2	2.9912	Single-prey	Odontocete	Gma	1.0519	0.7918	0.6401	0.4703	0.4703
<i>Globicephala melas</i>	2	3.0792	Single-prey	Odontocete	Gme	1.2829	0.8406	0.6335	0.4257	0.4257
<i>Grampus griseus</i>	2	2.5441	Single-prey	Odontocete	Gg	1.1941	0.8068	0.6328	0.4577	0.4577
<i>Megaptera novaeangliae</i>	1	4.4472	Filter	Rorqual	Mn	2.1839	2.0144	1.8621	1.6563	1.6563
<i>Mesoplodon densirostris</i>	5	2.9345	Single-prey	Odontocete	Md	0.4639	0.0020	-0.2024	-0.4092	-0.4092
<i>Orcinus orca</i>	2	3.4771	Single-prey	Odontocete	Oo	0.7544	0.6359	0.5393	0.4096	0.4096
<i>Phocoena phocoena</i>	3	1.5185	Single-prey	Odontocete	Pp	1.0350	0.8313	0.7372	0.6407	0.6407
<i>Physeter macrocephalus</i>	4	4.2856	Single-prey	Odontocete	Pm	0.7403	0.3781	0.1685	-0.0732	-0.0732
<i>Ziphius cavirostris</i>	5	3.4624	Single-prey	Odontocete	Zc	0.4286	-0.0079	-0.2229	-0.4496	-0.4496



8.1.3 Rearrange the row order in smydata to match smytree

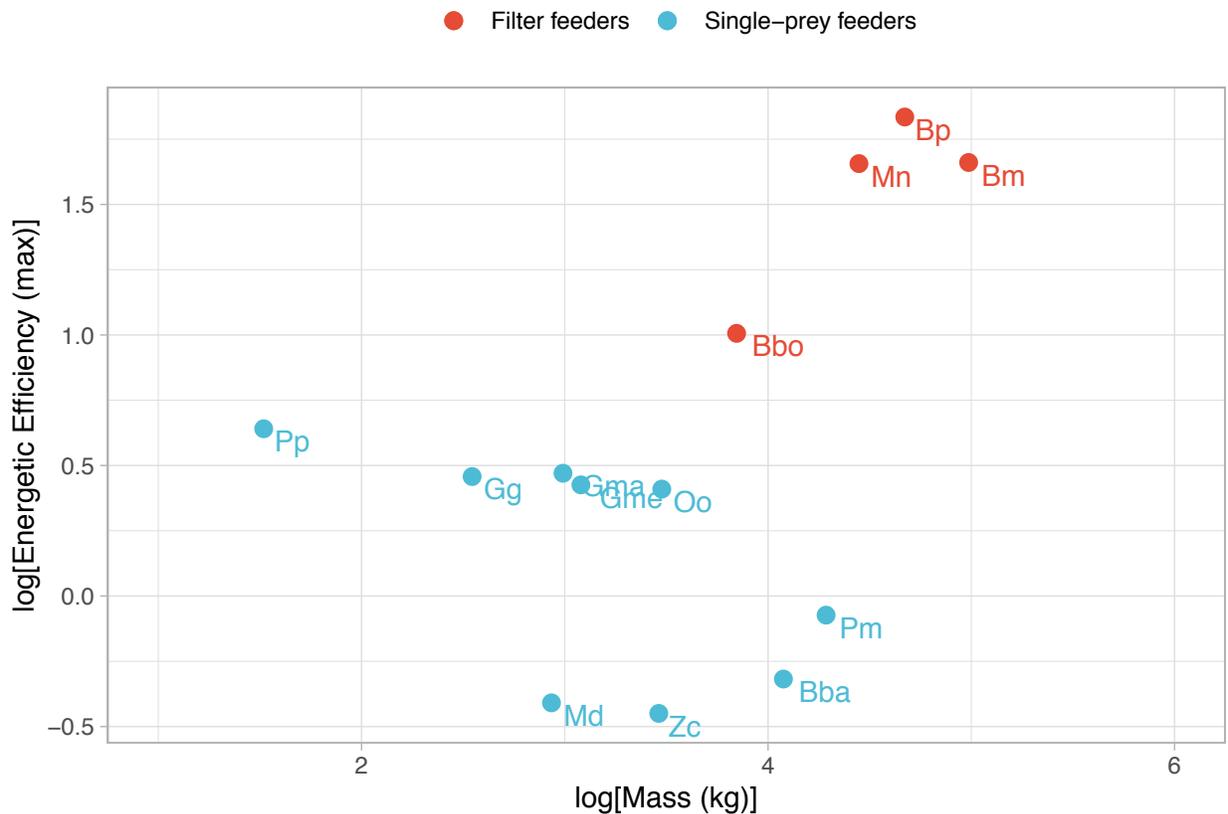
```
smydata <- smydata[match(smytree$tip.label,rownames(smydata)),]
rownames(smydata) <- NULL
kable(smydata,
      caption = "Model inputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
  kable_styling(latex_options = "scale_down")
```

```
rownames(smydata) <- smydata$species
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_75_smydata.rds")
```

```
saveRDS(smydata,fil)
```

8.2 Plot the data

```
ggplot(smydata, aes(x_mean, y = value, color = Group)) +  
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), shape = 16, size = 3,  
    aes(y = y_mean, color = "#4DBBD5FF")) +  
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), shape = 16, size = 3,  
    aes(y = y_mean, color = "#E64B35FF")) +  
  scale_color_manual(name = "",  
    values = c("#E64B35FF", "#4DBBD5FF"),  
    labels = c("Filter feeders", "Single-prey feeders")) +  
  theme_light() + theme(legend.position = "top") +  
  xlim(1,6) +  
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +  
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",  
    aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1) +  
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",  
    aes(y = y_mean, label = abbreviation), hjust = -0.3, vjust = 1.1)
```



8.3 Run OLS with feeding mode as a categorical predictor

8.3.1 Run OLS and model reduction using ML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "ML")
summary(m.ols)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 14.53054 17.35529 -2.265272
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.3743943 1.8678165 -0.7358294 0.4806
## fmSingle-prey  2.4993885 1.9287966 1.2958279 0.2273
## x_mean        0.6492673 0.4143996 1.5667665 0.1516
## fmSingle-prey:x_mean -0.9655002 0.4401019 -2.1938108 0.0559
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996 0.964
## fmSingle-prey:x_mean 0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -2.1047571 -0.5351318 0.4764041 0.6091823 1.3338155
##
## Residual standard error: 0.2880311
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.ols)
```

```
## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)  1 34.31832 0.0002
## fm          1 46.02248 0.0001
## x_mean      1 2.19528 0.1726
## fm:x_mean   1 4.81281 0.0559
```

```
m.ols.2 <- update(m.ols, ~ . - fm:x_mean)
anova(m.ols, m.ols.2)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## m.ols      1  5 14.53055 17.35529 -2.265272
## m.ols.2    2  4 18.09937 20.35917 -5.049688 1 vs 2 5.56883 0.0183
```

8.3.1.1 Compare to an intercept-only model

```
m.ols.0 <- gls(y_mean ~ 1, data = smydata, method = "ML")
anova(m.ols, m.ols.0)
```

```
##      Model df      AIC      BIC    logLik  Test L.Ratio p-value
## m.ols      1  5 14.53054 17.35529 -2.265272
## m.ols.0    2  2 33.62578 34.75568 -14.812891 1 vs 2 25.09524 <.0001
```

```
m.ols.p <- anova(m.ols, m.ols.0)$`p-value`[2]
```

8.3.2 Estimate final model using REML

```
m.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
summary(m.ols)
```

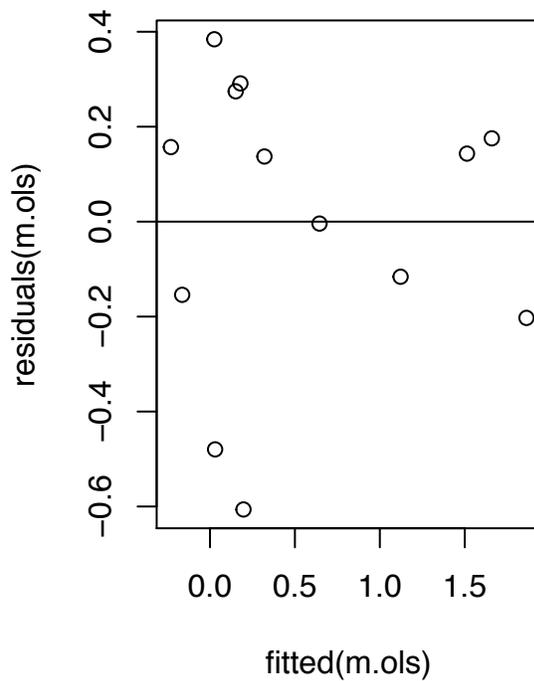
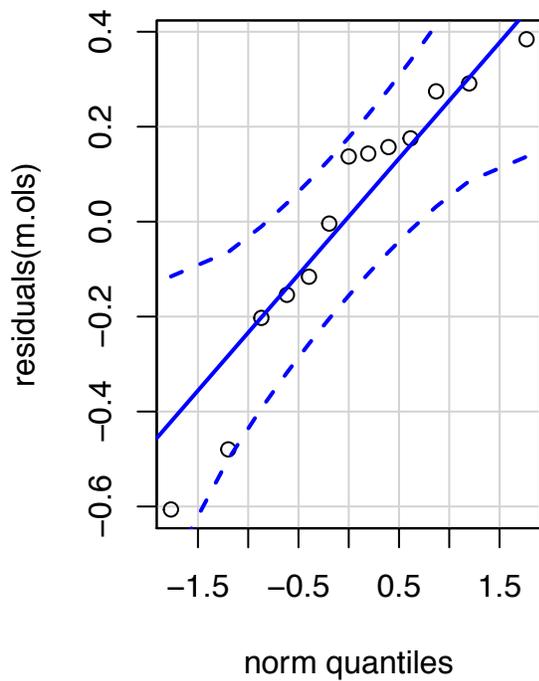
```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 21.36654 22.35266 -5.683268
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.3743943  1.8678165  -0.7358294  0.4806
## fmSingle-prey  2.4993885  1.9287966  1.2958279  0.2273
## x_mean        0.6492673  0.4143996  1.5667665  0.1516
## fmSingle-prey:x_mean -0.9655002  0.4401019 -2.1938108  0.0559
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.968
## x_mean         -0.996  0.964
## fmSingle-prey:x_mean  0.938 -0.989 -0.942
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7512638 -0.4452566  0.3963922  0.5068704  1.1098016
##
## Residual standard error: 0.3461703
## Degrees of freedom: 13 total; 9 residual
```

```
m.ols.param <- as.data.frame(t(summary(m.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.ols.param <- m.ols.param[5:8]
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pgls\\",
             "Figure4_75_m_ols_param.rds")
saveRDS(m.ols.param,fil)
```

8.3.2.1 Model diagnostics

8.3.2.1.1 QQ-plot and Residuals vs fitted plot

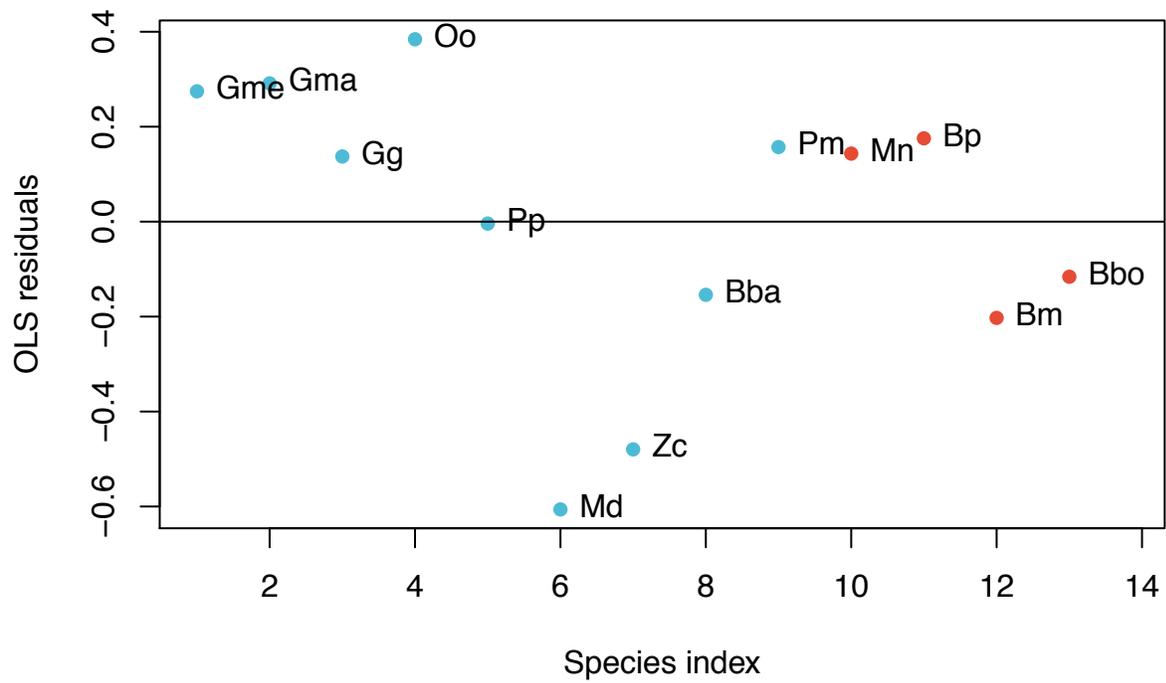
```
par(mfrow=c(1,2))
qqPlot(residuals(m.ols), id=FALSE)
plot(fitted(m.ols), residuals(m.ols))
abline(0,0)
```



8.3.3 Evaluate for phylogenetic correlation

8.3.3.1 Plot residuals ordered “by phylogeny” (i.e. in the order of tips of the phylogenetic tree)

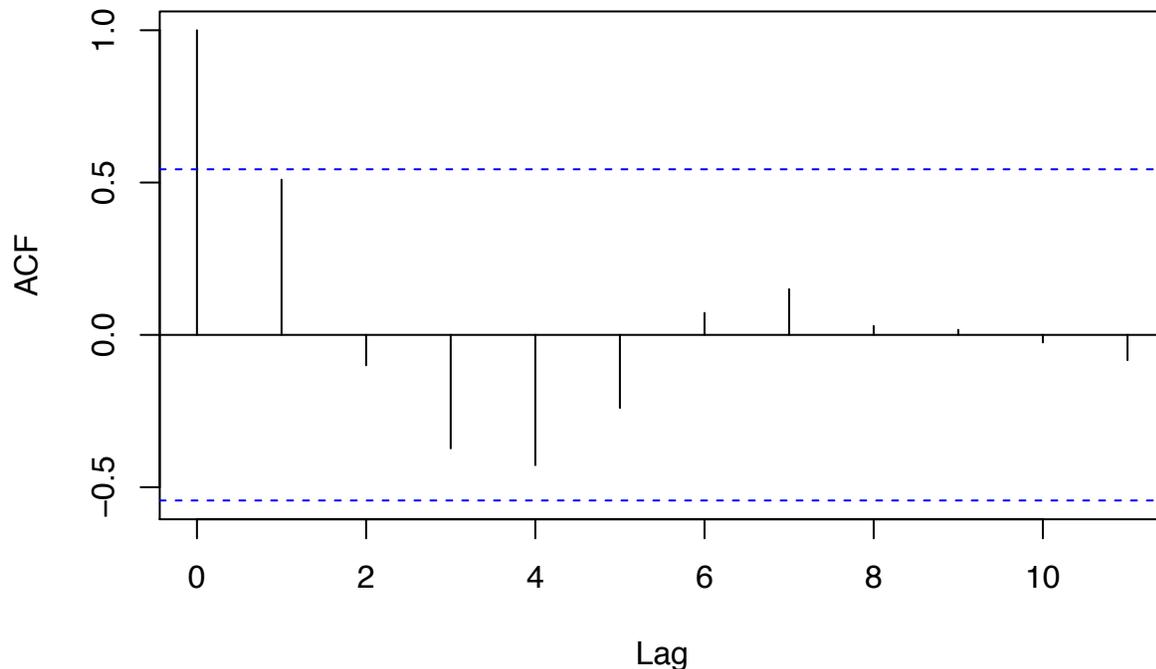
```
is_tip <- smytree$edge[,2]<=length(smytree$tip.label)
ordered_tips <- smytree$edge[is_tip,2]
oj <- residuals(m.ols)
tl <- smytree$tip.label[ordered_tips]
res <- oj[tl]
plot(oj[tl], pch=16, ylab="OLS residuals", xlab="Species index",
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata[tl,"fm"])],
     xlim=c(1,13.8))
abline(0,0)
text(oj[tl], labels=abbreviation[tl], pos=4)
```



8.3.3.2 Plot autocorrelation function of residuals ordered “by phylogeny”

```
acf(res, main="Series: residuals sorted by phylogeny")
```

Series: residuals sorted by phylogeny



8.4 Run a pGLS with feeding mode as a categorical predictor

8.4.1 Estimate Pagel's λ (amount of phylogenetic signal) for each trait separately

```
lambdax <- phylosig(smytree, smydata$x_mean, method = "lambda", test = T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
lambday <- phylosig(smytree, smydata$y_mean, method = "lambda", test = T)
## [1] "x has no names; assuming x is in the same order as tree$tip.label"
cbind(lambdax, lambday)

##      lambdax      lambday
## lambda 1.014327  1.015473
## logL   -12.69024  -5.450138
## logL0  -17.41681  -14.81289
## P       0.002107892 1.509499e-05
```

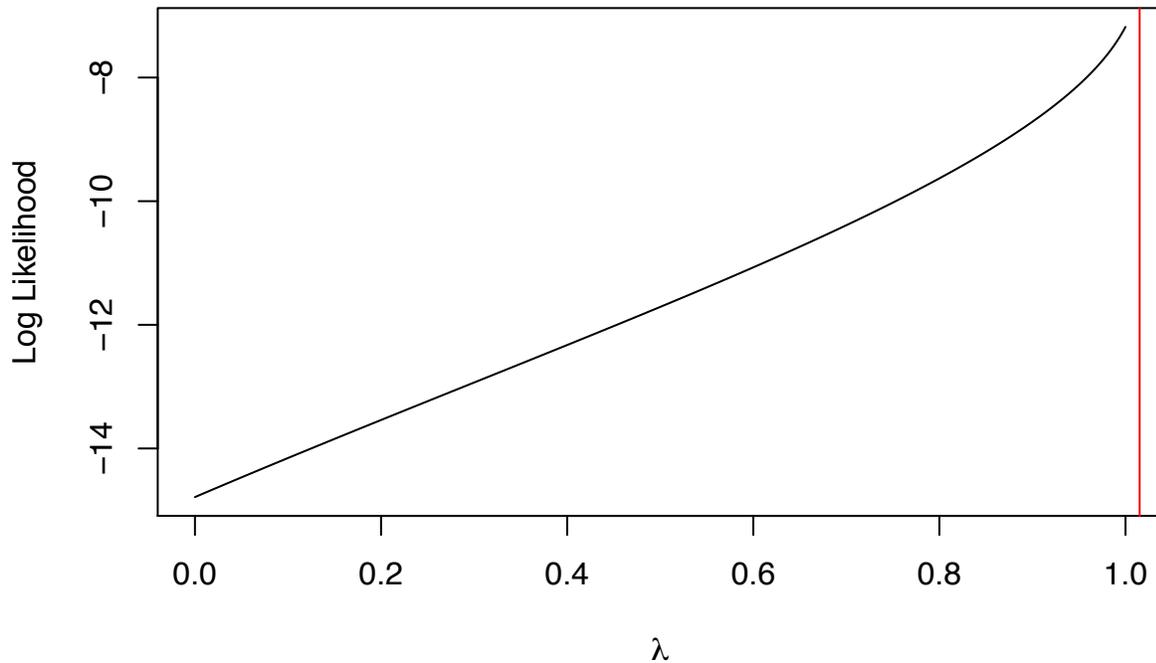
8.4.2 Plot likelihood surface for Pagel's λ for model without feeding mode as a covariate

λ estimate marked in red.

```
lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ x_mean, smydata,
  method = "REML", correlation = corPagel(value = lambda, phy = smytree,
    fixed = TRUE))))
```

```
plot(lik ~ lambda, type = "l", main =
      expression(paste("Prey energy to body mass Likelihood Plot for ", lambda)),
      ylab = "Log Likelihood", xlab = expression(lambda))
m.pa.only <- gls(y_mean ~ x_mean, data = smydata, correlation =
                 corPagel(value = 0, phy = smytree, fixed = FALSE), method = "REML")
abline(v = m.pa.only$modelStruct[1], col = "red")
```

Prey energy to body mass Likelihood Plot for λ



8.4.3 Estimate Pagel's λ using REML

If λ is estimated to be greater than 1, fix it at 1, if smaller than 0, fix it at 0.

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
                  corPagel(1, phy = smytree, fixed = FALSE), method = "REML")
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC    logLik
## 11.65858 12.84193 0.1707077
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
## 1.015264
##
```

```
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)  -1.1188592 1.1444009 -0.9776811  0.3538
## fmSingle-prey  1.8286864 1.2186261  1.5006131  0.1677
## x_mean        0.5840682 0.2502383  2.3340478  0.0445
## fmSingle-prey:x_mean -0.7944905 0.2736602 -2.9032007  0.0175
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey  -0.939
## x_mean         -0.965  0.906
## fmSingle-prey:x_mean  0.882 -0.952 -0.914
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.4537284 -0.3859101  0.5151334  0.8208066  1.2504427
##
## Residual standard error: 0.3450228
## Degrees of freedom: 13 total; 9 residual
```

```
anova(m.pgls.nlme)
```

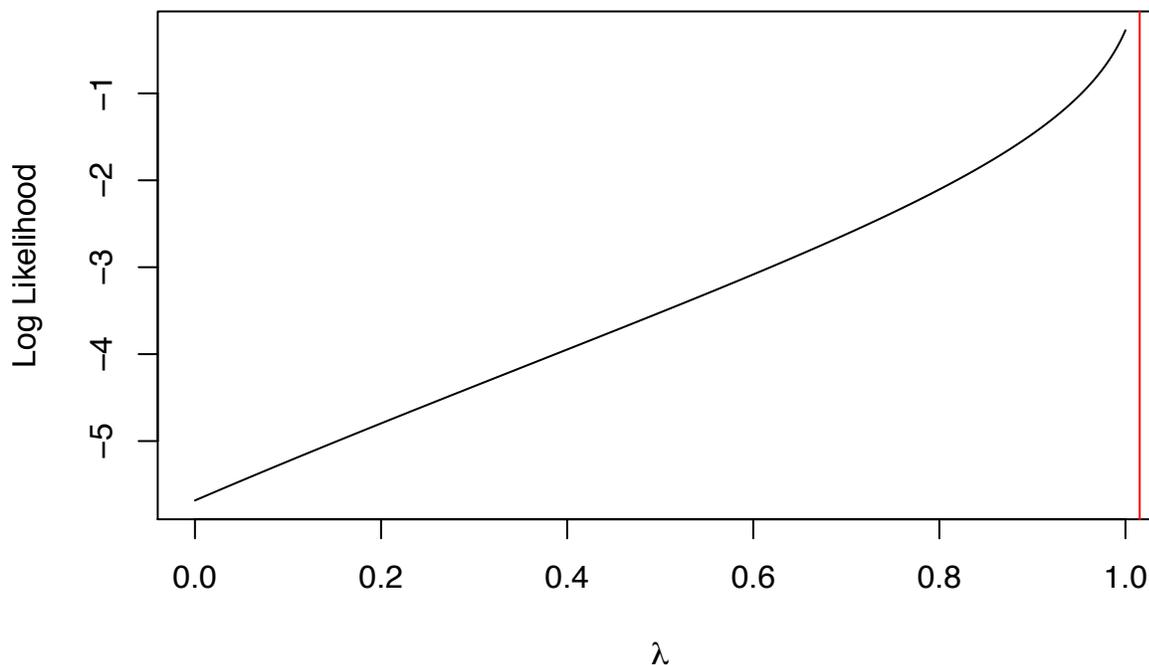
```
## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)    1  6.718898  0.0291
## fm              1 16.683007  0.0027
## x_mean         1  0.627603  0.4486
## fm:x_mean      1  8.428575  0.0175
```

```
lambda.est <- as.numeric(m.pgls.nlme$modelStruct[1])
if(lambda.est > 1){lambda.est <- 1} else if(lambda.est < 0){lambda.est <- 0}
```

8.4.4 Plot likelihood surface for Pagel's λ - our estimate marked in red

```
lambda <- seq(0, 1, length.out = 500)
lik <- sapply(lambda, function(lambda) logLik(gls(y_mean ~ fm * x_mean, smydata,
                                                method = "REML", correlation =
                                                corPagel(value = lambda, phy = smytree, fixed = TRUE))))
plot(lik ~ lambda, type = "l", main =
     expression(paste("Energetic Efficiency to Body mass Likelihood Plot for ", lambda)),
     ylab = "Log Likelihood", xlab = expression(lambda)),
abline(v = m.pgls.nlme$modelStruct, col = "red")
```

Energetic Efficiency to Body mass Likelihood Plot for λ



8.4.5 Run pGLS and model reduction with a fixed Pagel's λ (using ML)

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =  
  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")  
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by maximum likelihood  
## Model: y_mean ~ fm * x_mean  
## Data: smydata  
##      AIC      BIC   logLik  
## 3.951803 6.77655 3.024098  
##  
## Correlation Structure: corPagel  
## Formula: ~1  
## Parameter estimate(s):  
## lambda  
##      1  
##  
## Coefficients:  
##              Value Std.Error   t-value p-value  
## (Intercept)   -1.1282985 1.0996406 -1.026061 0.3316  
## fmSingle-prey  1.7333640 1.1765603  1.473247 0.1748  
## x_mean         0.5865165 0.2406442  2.437277 0.0375  
## fmSingle-prey:x_mean -0.7657523 0.2655301 -2.883863 0.0181  
##  
## Correlation:
```

```

##          (Intr) fmSng- x_mean
## fmSingle-prey      -0.935
## x_mean             -0.966  0.903
## fmSingle-prey:x_mean 0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7940569 -0.4993482 0.6476509 1.1338338 1.5715267
##
## Residual standard error: 0.27219
## Degrees of freedom: 13 total; 9 residual
anova(m.pgls.nlme)

## Denom. DF: 9
##      numDF  F-value p-value
## (Intercept)  1  7.600033 0.0222
## fm           1 18.786745 0.0019
## x_mean       1  0.173982 0.6864
## fm:x_mean    1  8.316663 0.0181

m.pgls.nlme.2 <- update(m.pgls.nlme, ~ . - fm:x_mean)
anova(m.pgls.nlme, m.pgls.nlme.2)

##      Model df      AIC      BIC  logLik  Test L.Ratio p-value
## m.pgls.nlme      1  5  3.951803  6.77655  3.024098
## m.pgls.nlme.2    2  4 10.459584 12.71938 -1.229792 1 vs 2 8.50778 0.0035

m.pgls.fm <- gls(y_mean ~ fm, data = smydata, correlation =
                 corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "ML")
summary(m.pgls.fm)

## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm
## Data: smydata
##      AIC      BIC  logLik
##  8.589544 10.28439 -1.294772
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##      Value Std.Error  t-value p-value
## (Intercept)  1.460672 0.3585642  4.073669 0.0018
## fmSingle-prey -1.458801 0.4244015 -3.437314 0.0056
##
## Correlation:
##      (Intr)
## fmSingle-prey -0.845
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.1980625 -0.8428870 0.5269103 1.1170579 1.6835824

```

```
##
## Residual standard error: 0.3794493
## Degrees of freedom: 13 total; 11 residual
```

```
summary(m.pgls.nlme)
```

```
## Generalized least squares fit by maximum likelihood
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC   logLik
## 3.951803 6.77655 3.024098
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##              Value Std.Error  t-value p-value
## (Intercept) -1.1282985 1.0996406 -1.026061 0.3316
## fmSingle-prey 1.7333640 1.1765603 1.473247 0.1748
## x_mean      0.5865165 0.2406442 2.437277 0.0375
## fmSingle-prey:x_mean -0.7657523 0.2655301 -2.883863 0.0181
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey -0.935
## x_mean      -0.966 0.903
## fmSingle-prey:x_mean 0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7940569 -0.4993482 0.6476509 1.1338338 1.5715267
##
## Residual standard error: 0.27219
## Degrees of freedom: 13 total; 9 residual
```

8.4.5.1 Compare to an intercept-only model

```
m.pgls.nlme.0 <- gls(y_mean ~ 1, smydata, correlation = corPagel(value = lambda.est,
  phy = smytree, fixed = TRUE), method = "ML")
anova(m.pgls.nlme, m.pgls.nlme.0)
```

```
##           Model df      AIC      BIC   logLik  Test  L.Ratio
## m.pgls.nlme      1  5  3.951803  6.77655  3.024098
## m.pgls.nlme.0    2  2 16.073415 17.20331 -6.036707 1 vs 2 18.12161
##
##           p-value
## m.pgls.nlme
## m.pgls.nlme.0 4e-04
```

```
m.pgls.p <- anova(m.pgls.nlme, m.pgls.nlme.0)$`p-value`[2]
```

8.4.6 Estimate final model using REML

```
m.pgls.nlme <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
                  corPagel(lambda.est, phy = smytree, fixed = TRUE), method = "REML")
summary(m.pgls.nlme)
```

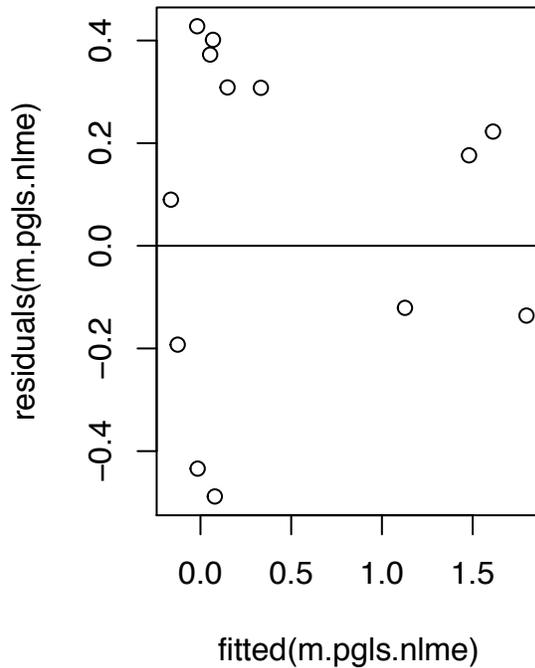
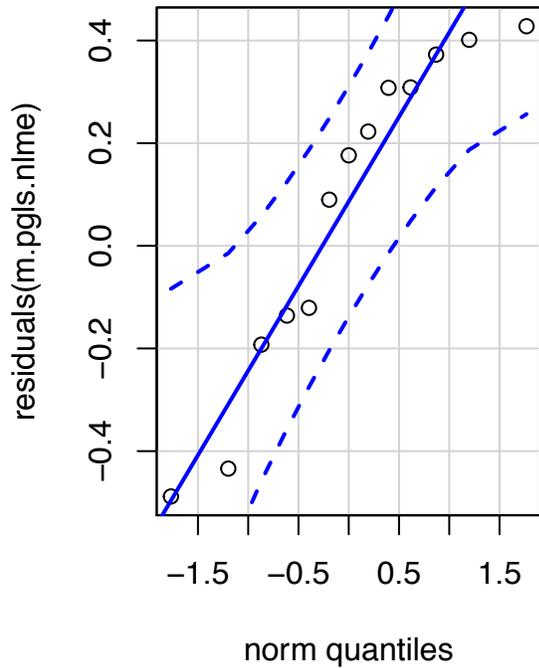
```
## Generalized least squares fit by REML
## Model: y_mean ~ fm * x_mean
## Data: smydata
##      AIC      BIC      logLik
## 10.54799 11.53411 -0.2739939
##
## Correlation Structure: corPagel
## Formula: ~1
## Parameter estimate(s):
## lambda
##      1
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept)   -1.1282985 1.0996406 -1.026061 0.3316
## fmSingle-prey  1.7333640 1.1765603  1.473247 0.1748
## x_mean         0.5865165 0.2406442  2.437277 0.0375
## fmSingle-prey:x_mean -0.7657523 0.2655301 -2.883863 0.0181
##
## Correlation:
##              (Intr) fmSng- x_mean
## fmSingle-prey   -0.935
## x_mean          -0.966  0.903
## fmSingle-prey:x_mean 0.875 -0.954 -0.906
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.4927456 -0.4154828 0.5388781 0.9434067 1.3075893
##
## Residual standard error: 0.3271316
## Degrees of freedom: 13 total; 9 residual
```

```
m.pgls.param <- as.data.frame(t(summary(m.pgls.nlme)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.pgls.param <- m.pgls.param[5:8]
```

8.4.6.1 Model diagnostics

8.4.6.1.1 QQ-plot and Residuals vs fitted plot

```
par(mfrow = c(1,2))
qqPlot(residuals(m.pgls.nlme), id = FALSE)
plot(fitted(m.pgls.nlme), residuals(m.pgls.nlme))
abline(0,0)
```



8.5 Estimate confidence intervals by bootstrapping

8.5.1 Bootstrap and compute percentile confidence intervals

```
d_sub <- filter(d_full, MR.exponent==.75)
index <- d_sub %>% group_by(Spec) %>% summarize(ix=length(y))
index # number of prey categories for each species
```

```
## # A tibble: 13 x 2
##   Spec                ix
##   <fct>              <int>
## 1 Balaenoptera_bonaerensis    5
## 2 Balaenoptera_musculus      7
## 3 Balaenoptera_physalus      7
## 4 Berardius_bairdii         19
## 5 Globicephala_macrorhynchus 12
## 6 Globicephala_melas       12
## 7 Grampus_griseus           5
## 8 Megaptera_novaeangliae     8
## 9 Mesoplodon_densirostris    3
## 10 Orcinus_orca              12
## 11 Phocoena_phocoena         5
## 12 Physeter_macrocephalus    18
## 13 Ziphius_cavirostris       16
```

```

smydata.orig <- smydata
y_mean <- by(d_sub, d_sub$Spec, with, weighted.mean(y, Percent))

spec <- unique(d_sub$Spec)
spec <- spec[match(spec, smydata$species)]

runpGls <- function(smydata, smytree){
  out <- tryCatch(
    {
      model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
        corPagel(lambda.est, phy = smytree, fixed = FALSE),
        method = "REML")
      as.numeric(model.pgls$modelStruct[1])
    },
    error=function(cond) {
      return(NA)
    }
  )
}

a.ols <- matrix(nrow=10000, ncol=4)
a.pgls <- matrix(nrow=10000, ncol=4)
b <- matrix(nrow=10000, ncol=length(spec))
boot.lambdas <- rep(NA, 10000)
for(i in 1:10000){
  for(sp in 1:length(spec)){
    ix <- sample(1:index$ix[index$Spec==spec[sp]], replace = T)
    y_mean[sp] <- sum(d_sub[d_sub$Spec==spec[sp], "y"][ix] *
      d_sub[d_sub$Spec==spec[sp], "Percent"][ix]) /
      sum(d_sub[d_sub$Spec==spec[sp], "Percent"][ix])
  }
  smydata$y_mean <- y_mean

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata, method = "REML")
  myout <- runpGls(smydata, smytree)
  boot.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(lambda.est, phy = smytree, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata, correlation =
      corPagel(l.est, phy = smytree, fixed = TRUE), method = "REML")
  }

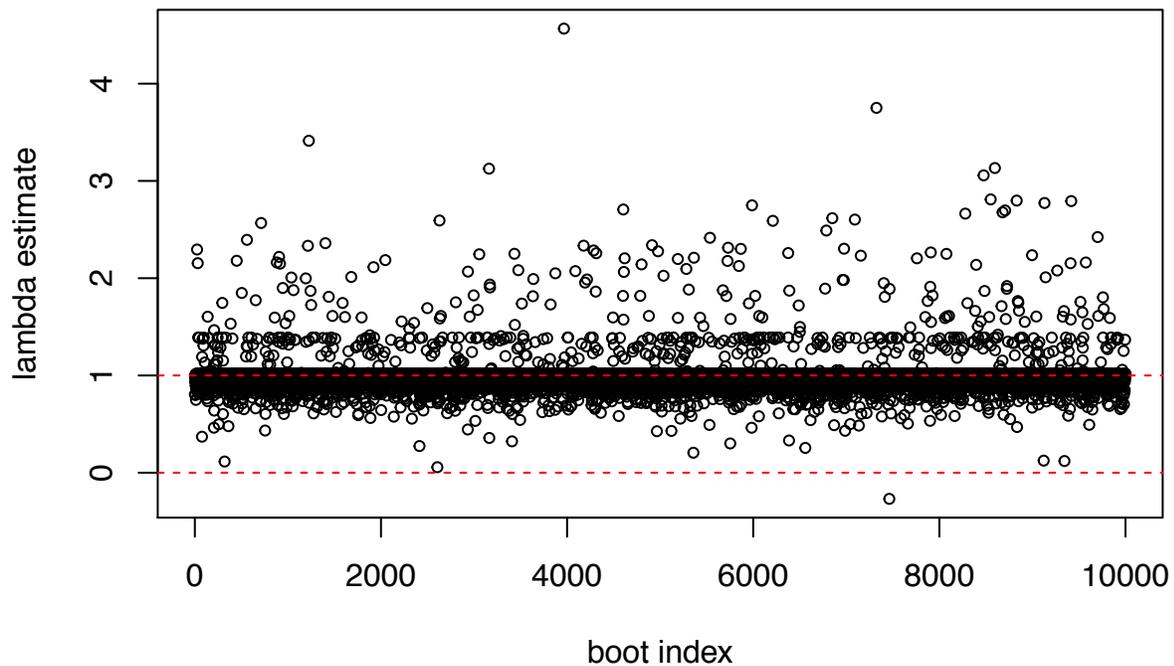
  a.ols[i,] <- c(coef(model.ols)[1], coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3], coef(model.ols)[3]+coef(model.ols)[4])
  a.pgls[i,] <- c(coef(model.pgls)[1], coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3], coef(model.pgls)[3]+coef(model.pgls)[4])
  b[i,] <- predict(model.ols)
}

```

```
# number of pGLS models, where lambda could not be estimated ==> used original value:  
sum(is.na(boot.lambdas))
```

```
## [1] 216
```

```
plot(boot.lambdas, cex=.7, xlab="boot index", ylab="lambda estimate")  
abline(h=0,lty="dashed",col="red")  
abline(h=1,lty="dashed",col="red")
```



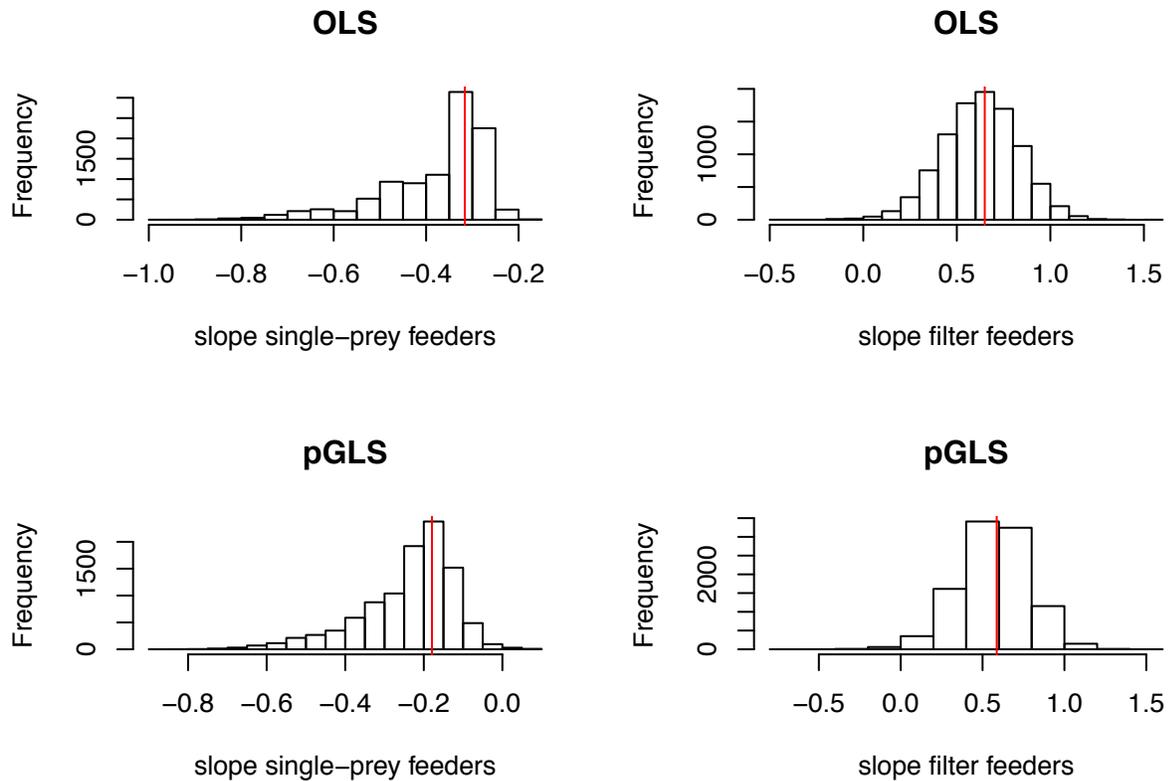
```
preds <- apply(b, 2, quantile, c(0.025, 0.975))  
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\ppls\\",  
             "Figure4_75_bootstrap_b.rds")  
saveRDS(b,fil)  
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\ppls\\",  
             "Figure4_75_bootstrap_preds.rds")  
saveRDS(preds,fil)  
  
df.boot.ols <- data.frame(cbind(t(m.ols.param),t(t(apply(a.ols, 2, mean))),  
                             t(apply(a.ols, 2, quantile, c(0.025, 0.975))))))  
names(df.boot.ols) <- c("obs","bootest","lowerCI","upperCI")  
df.boot.ppls <- data.frame(cbind(t(m.ppls.param),t(t(apply(a.ppls, 2, mean))),  
                             t(apply(a.ppls, 2, quantile, c(0.025, 0.975))))))  
names(df.boot.ppls) <- c("obs","bootest","lowerCI","upperCI")  
  
par(mfrow=c(2,2))  
hist(a.ols[,4], xlab="slope single-prey feeders", main="OLS")
```

```

abline(v=m.ols.param[4], col="red")
hist(a.ols[,3], xlab="slope filter feeders", main="OLS")
abline(v=m.ols.param[3], col="red")

hist(a.pglsl[,4], xlab="slope single-prey feeders", main="pGLS")
abline(v=m.pglsl.param[4], col="red")
hist(a.pglsl[,3], xlab="slope filter feeders", main="pGLS")
abline(v=m.pglsl.param[3], col="red")

```



8.5.2 Compute BCa (bias-corrected and accelerated) confidence intervals

```

smydata <- smydata.orig

# compute bias-correction factor from the proportion of bootstrap estimates
# that are less than the observed estimate

bootBC <- function(bootEst, Est){
  B <- ncol(bootEst)*nrow(bootEst) # number of bootstrap samples
  propLess <- sum(bootEst < Est)/B # proportion of replicates less than observed stat
  z0 <- qnorm(propLess) # bias correction
  return(z0)
}

z0.ols <- numeric()
for (i in 1:ncol(a.ols)){

```

```

z0.ols[i] <- bootBC(t(t(a.ols[,i])),as.numeric(m.ols.param[i]))
}

z0.pgls <- numeric()
for (i in 1:ncol(a.pgls)){
z0.pgls[i] <- bootBC(t(t(a.pgls[,i])),as.numeric(m.pgls.param[i]))
}

# compute acceleration factor, which is related to the skewness of bootstrap estimates.
# Use jackknife replicates to estimate.

jStat.ols <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jStat.pgls <- matrix(ncol=nrow(smydata),nrow=4) # jackknife replicates
jack.lambdas <- rep(NA,nrow(smydata))
for (i in 1:nrow(smydata)) {
  d_sub <- subset(d_full, Spec==smydata$species[i] & MR.exponent==.75)
  y_mean.j <- numeric()
  for(j in 1:nrow(d_sub)){
    d_sub.j <- d_sub[-j,]
    y_mean.j[j] <- sum(d_sub.j$y*d_sub.j$Percent)/sum(d_sub.j$Percent)
  }
  smydata.j <- smydata
  smydata.j$y_mean[i] <- mean(y_mean.j)
  pruned.tree <- drop.tip(smytree,smytree$tip.label[-match(smydata.j$species,
    smytree$tip.label)])

  smytree.j <- pruned.tree
  smydata.j <- smydata.j[match(smytree.j$tip.label,rownames(smydata.j)),]

  model.ols <- gls(y_mean ~ fm * x_mean, data = smydata.j, method = "REML")

  myout <- runpGls(smydata.j,smytree.j)
  jack.lambdas[i] <- myout

  if(is.na(myout)){
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(lambda.est, phy = smytree.j, fixed = TRUE),
      method = "REML")
  } else {
    if(myout > 1){l.est <- 1} else if(myout < 0){l.est <- 0} else {l.est <- myout}
    model.pgls <- gls(y_mean ~ fm * x_mean, data = smydata.j, correlation =
      corPagel(l.est, phy = smytree.j, fixed = TRUE), method = "REML")
  }

  jStat.ols[,i] <- as.numeric(c(coef(model.ols)[1],coef(model.ols)[1]+coef(model.ols)[2],
    coef(model.ols)[3],coef(model.ols)[3]+coef(model.ols)[4]))
  jStat.pgls[,i] <- as.numeric(c(coef(model.pgls)[1],
    coef(model.pgls)[1]+coef(model.pgls)[2],
    coef(model.pgls)[3],
    coef(model.pgls)[3]+coef(model.pgls)[4]))
}
jackEst.ols <- t(t(apply(jStat.ols, 1, mean))) # jackknife estimate

```

```

jackEst.pgls <- t(t(apply(jStat.pgls, 1, mean))) # jackknife estimate
jack.lambdas # lambdas of the jackknifed models

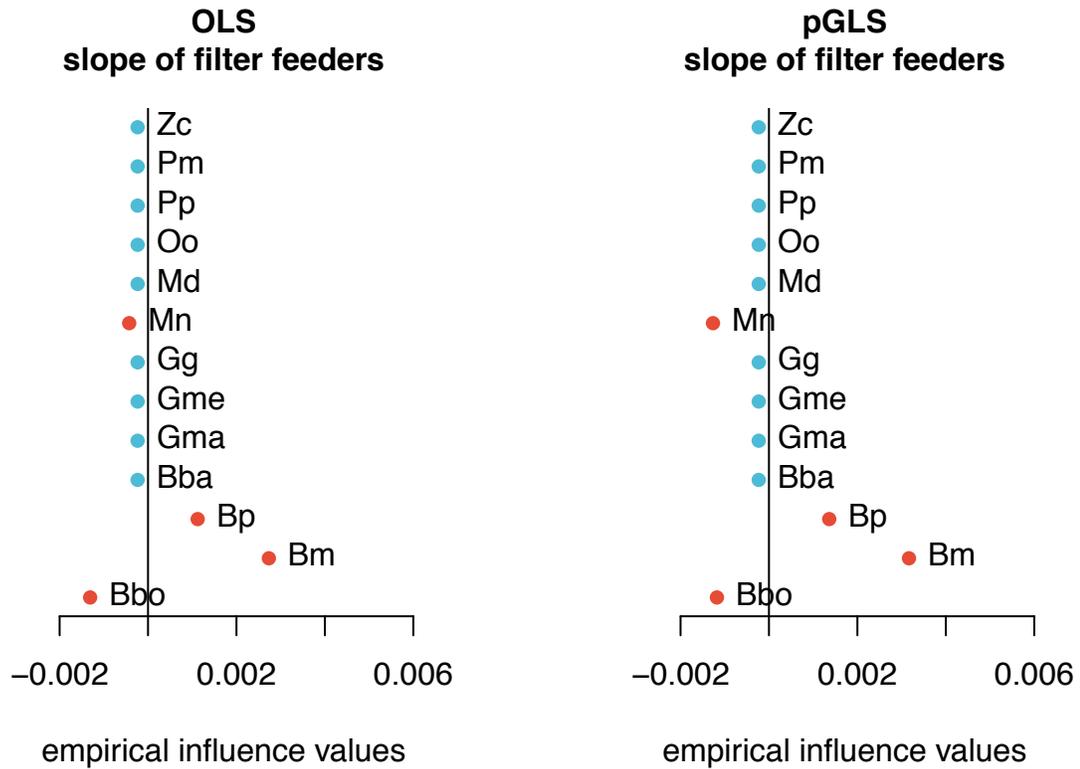
## [1] 1.015266 1.015282 1.015242 1.015270 1.015081 1.015438 1.015284
## [8] 1.015255 1.015246 1.015266 1.015835 1.015263 1.015261

num.ols <- numeric(); den.ols <- numeric(); ahat.ols <- numeric()
num.pgls <- numeric(); den.pgls <- numeric(); ahat.pgls <- numeric()
for (i in 1:nrow(jStat.ols)){
  num.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^3 )
  den.ols[i] <- sum( (jackEst.ols[i,]-jStat.ols[i,])^2 )
  ahat.ols[i] <- num.ols[i]/(6*den.ols[i]^(3/2)) # ahat based on jackknife
  num.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^3 )
  den.pgls[i] <- sum( (jackEst.pgls[i,]-jStat.pgls[i,])^2 )
  ahat.pgls[i] <- num.pgls[i]/(6*den.pgls[i]^(3/2)) # ahat based on jackknife
}

# influential species:
par(mfrow=c(1,2))
plot(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("OLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[3,]-jStat.ols[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[3,]-jStat.ols[3,])-0.001,
            max(jackEst.ols[3,]-jStat.ols[3,])+0.003), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[3,]-jStat.ols[3,])-0.001)*1e3)/1e3,
                   round((max(jackEst.ols[3,]-jStat.ols[3,])+0.003)*1e3)/1e3, 2e-3))
title(paste0("pGLS", "\nslope of filter feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgls[3,]-jStat.pgls[3,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```

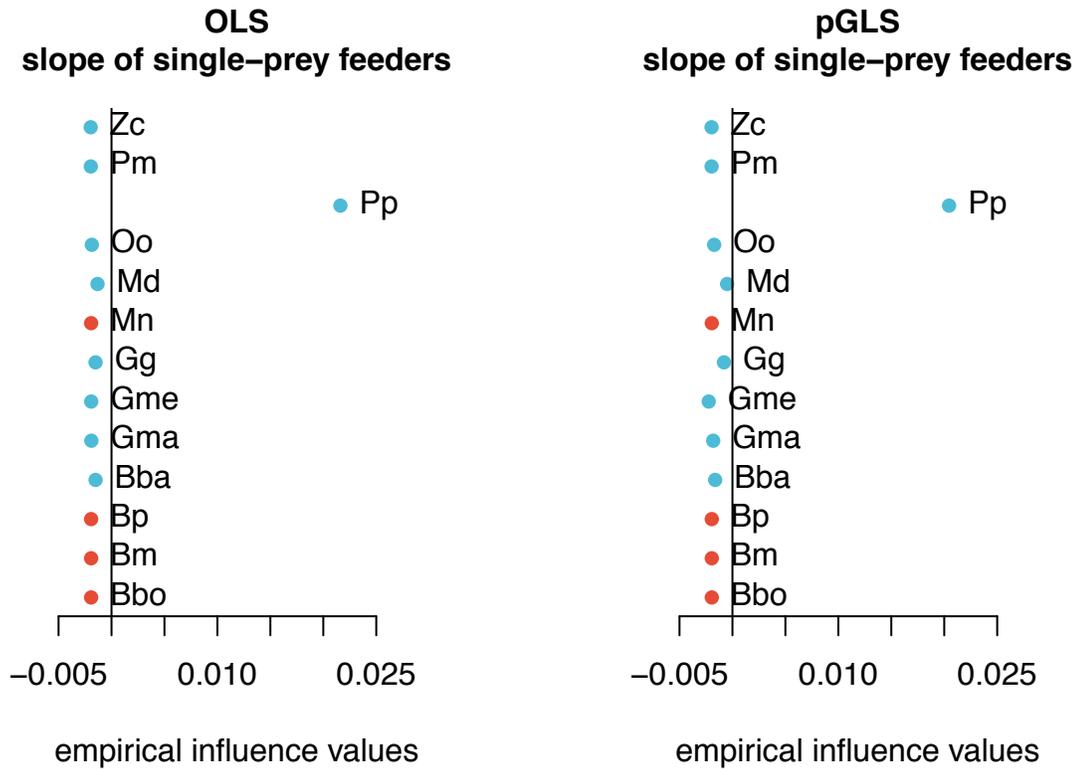


```

par(mfrow=c(1,2))
plot(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("OLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.ols[4,]-jStat.ols[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

plot(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)), pch=16,
     col=c("#E64B35FF", "#4DBBD5FF")[as.numeric(smydata$fm)],
     xlab="empirical influence values", ylab="",
     xlim=c(min(jackEst.ols[4,]-jStat.ols[4,])-0.003,
            max(jackEst.ols[4,]-jStat.ols[4,])+0.007), axes=F)
axis(side=1, at=seq(round((min(jackEst.ols[4,]-jStat.ols[4,])-0.003)*1e3)/1e3,
                    round((max(jackEst.ols[4,]-jStat.ols[4,])+0.007)*1e3)/1e3, 5e-3)))
title(paste0("pGLS", "\nslope of single-prey feeders"), line=1, cex.main=1)
abline(v=0)
text(jackEst.pgl[4,]-jStat.pgl[4,], (1:nrow(smydata)),
     labels=smydata$abbreviation, pos=4)

```



```
# adjust quantiles for 100*(1-alpha)% bootstrap BCa interval
```

```
alpha <- 0.05
zL.ols <- z0.ols + qnorm(alpha/2)
alpha1.ols <- pnorm(z0.ols + zL.ols / (1-ahat.ols*zL.ols))
zU.ols <- z0.ols + qnorm(1-alpha/2)
alpha2.ols <- pnorm(z0.ols + zU.ols / (1-ahat.ols*zU.ols))

zL.pgls <- z0.pgls + qnorm(alpha/2)
alpha1.pgls <- pnorm(z0.pgls + zL.pgls / (1-ahat.pgls*zL.pgls))
zU.pgls <- z0.pgls + qnorm(1-alpha/2)
alpha2.pgls <- pnorm(z0.pgls + zU.pgls / (1-ahat.pgls*zU.pgls))
```

```
cbind((alpha1.ols*100),(alpha2.ols*100)) # new quantiles OLS
```

```
##           [,1]      [,2]
## [1,]  0.564463502  93.92060
## [2,]  0.009036444  84.14969
## [3,]  6.525204235  99.52577
## [4,] 16.309403487  99.99228
```

```
cbind((alpha1.pgls*100),(alpha2.pgls*100)) # new quantiles pGLS
```

```
##           [,1]      [,2]
## [1,]  0.848125964  94.99382
## [2,]  0.004853031  82.03323
## [3,]  5.470157927  99.30962
```

```

## [4,] 18.610194494 99.99607
CI.ols <- matrix(nrow = ncol(a.ols), ncol=2)
for (i in 1:ncol(a.ols)){
  CI.ols[i,] <- quantile(a.ols[,i], c(alpha1.ols[i], alpha2.ols[i])) # BCa interval
}
df.boot.ols$lowerCIbca <- CI.ols[,1]
df.boot.ols$upperCIbca <- CI.ols[,2]

CI.pgls <- matrix(nrow = ncol(a.pgls), ncol=2)
for (i in 1:ncol(a.pgls)){
  CI.pgls[i,] <- quantile(a.pgls[,i], c(alpha1.pgls[i], alpha2.pgls[i])) # BCa interval
}
df.boot.pgls$lowerCIbca <- CI.pgls[,1]
df.boot.pgls$upperCIbca <- CI.pgls[,2]

```

8.5.3 Plot OLS model

```

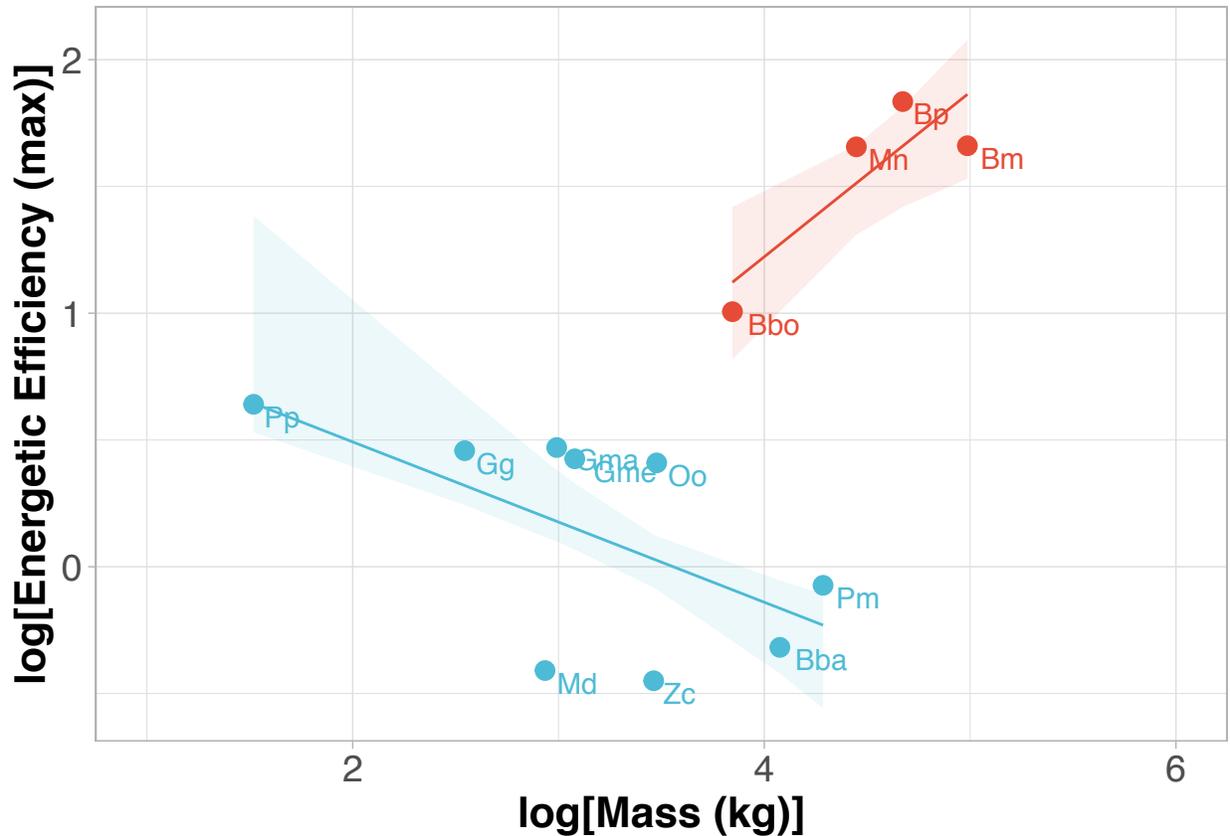
smydata <- smydata.orig

ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit,
                                       y_min = preds[1,], y_max = preds[2,]))

fig_ols <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
            linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
            linetype = 1) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Rorqual"), fill = "#E64B35FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  geom_ribbon(data = dplyr::filter(predframe2, Group == "Odontocete"), fill = "#4DBBD5FF",
            color = NA, alpha = 0.1, aes(ymin = y_min, ymax = y_max)) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +
  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_ols

```



8.5.3.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCI"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCI"],
  lowerCI.od = df.boot.ols["slope.od","lowerCI"],
  upperCI.od = df.boot.ols["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.ols["slope.rorq","obs"],
  slope.od = df.boot.ols["slope.od","obs"],
  lowerCI.rorq = df.boot.ols["slope.rorq","lowerCIbca"],
  upperCI.rorq = df.boot.ols["slope.rorq","upperCIbca"],
  lowerCI.od = df.boot.ols["slope.od","lowerCIbca"],
  upperCI.od = df.boot.ols["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.ols[,3],
  odontocete_slope=a.ols[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF"), color = NA, alpha = 0.3) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF"), color = NA, alpha = 0.3) +
  scale_fill_manual(name = "Feeding mode",
    values = c("#4DBBD5FF", "#E64B35FF"),
    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
    color = "#4DBBD5FF", linetype=1, size = 0.7) +

```

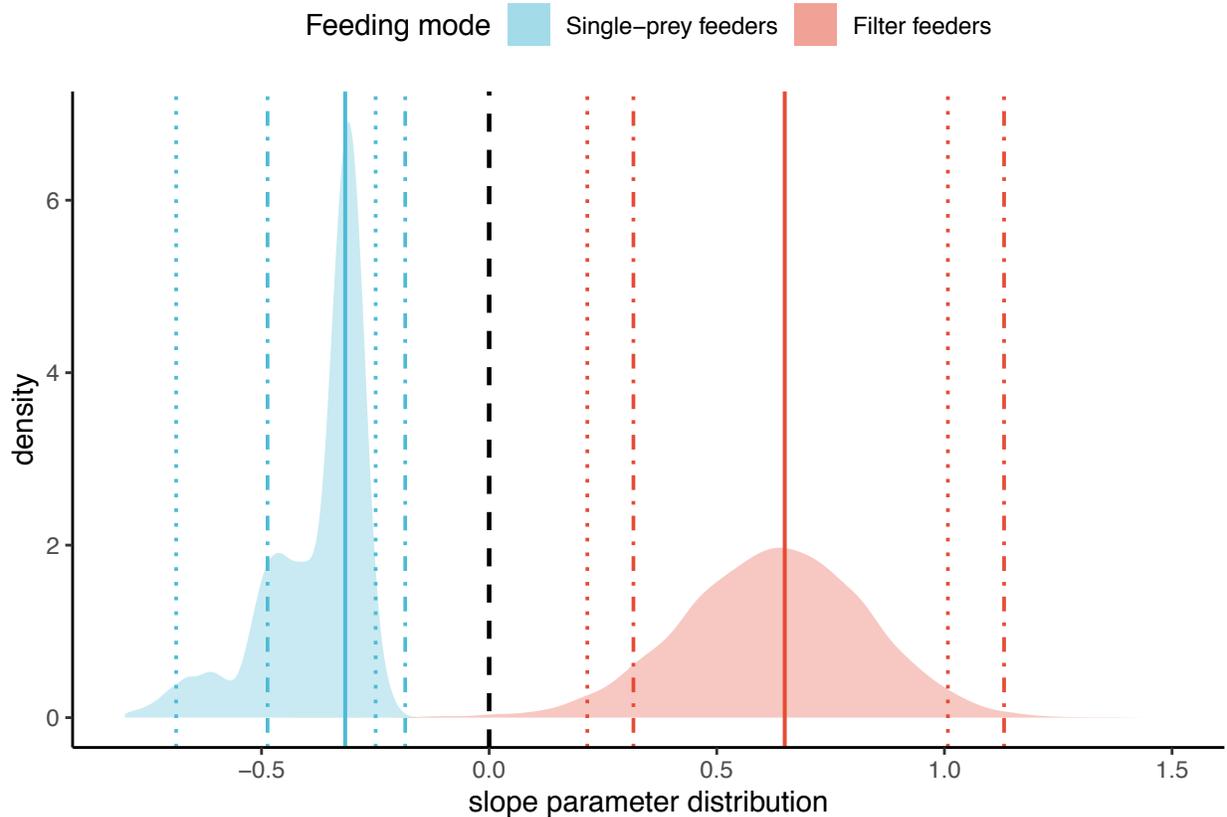
```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 43 rows containing non-finite values (stat_density).

Warning: Removed 1 rows containing non-finite values (stat_density).



```
rn <- rownames(df.boot.ols)
rownames(df.boot.ols) <- c("intercept filter","intercept single-prey",
                          "slope filter","slope single-prey")
knitr::kable(df.boot.ols,
             caption = "OLS 95%\% Bootstrap Pctl and BCa CI",
             format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
  kable_styling(latex_options = "hold_position")
```

Table 21: OLS 95% Bootstrap Pctl and BCa CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-1.3744	-1.2867	-3.0244	0.5363	-3.4994	0.1281
intercept single-prey	1.1250	1.3439	0.9186	2.4295	0.6588	1.7316
slope filter	0.6493	0.6257	0.2156	1.0076	0.3169	1.1311
slope single-prey	-0.3162	-0.3797	-0.6875	-0.2494	-0.4868	-0.1843

```
rownames(df.boot.ols) <- rn
```

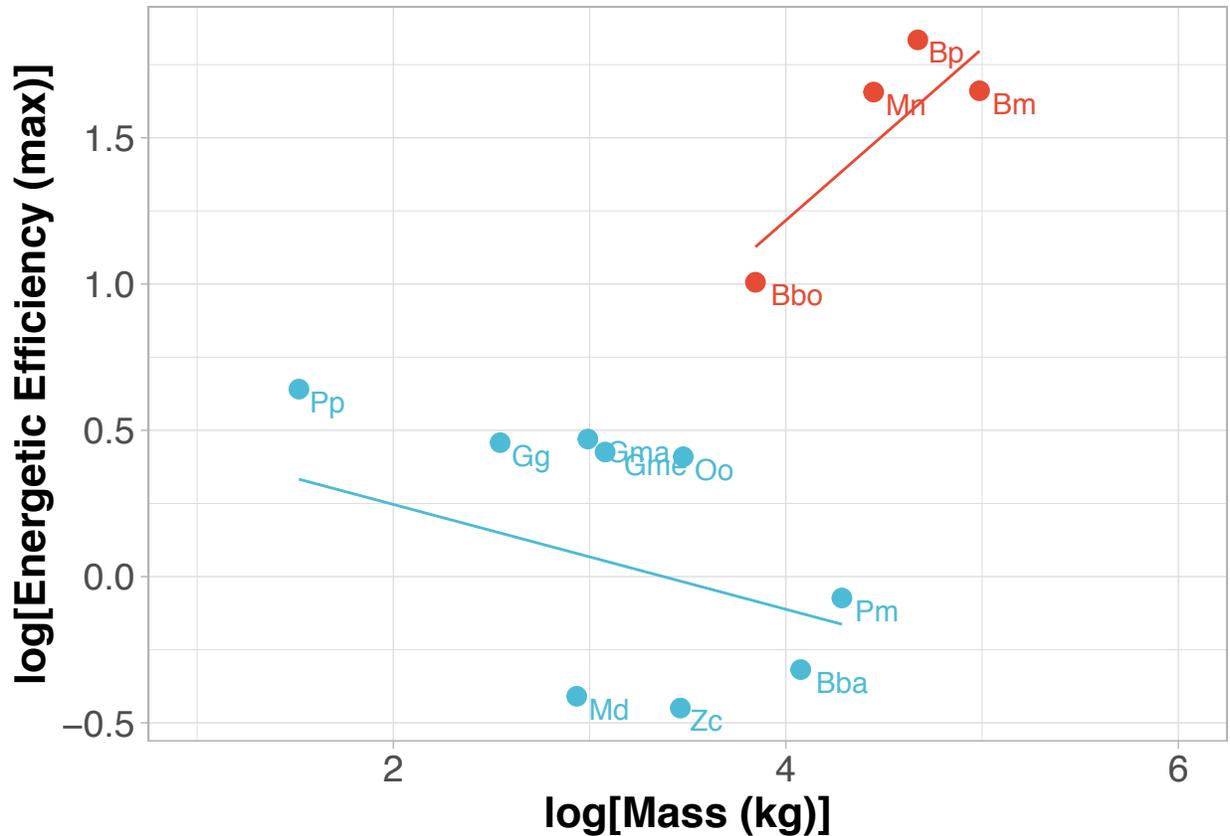
8.5.4 Plot pGLS model

```
pgls.fit <- predict(m.pgls.nlme)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))

fig_pgls <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
            linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
            linetype = 1) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
                                                                    face = "bold")) +

  xlim(1,6) +
  labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]") +
  geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1) +
  geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
            aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

fig_pgls
```



8.5.4.1 Plot kernel density distributions of slopes

```

model_param <- data.frame(slope.rorq = df.boot.pglsl["slope.rorq","obs"],
                          slope.od = df.boot.pglsl["slope.od","obs"],
                          lowerCI.rorq = df.boot.pglsl["slope.rorq","lowerCI"],
                          upperCI.rorq = df.boot.pglsl["slope.rorq","upperCI"],
                          lowerCI.od = df.boot.pglsl["slope.od","lowerCI"],
                          upperCI.od = df.boot.pglsl["slope.od","upperCI"])
model_param_bca <- data.frame(slope.rorq = df.boot.pglsl["slope.rorq","obs"],
                              slope.od = df.boot.pglsl["slope.od","obs"],
                              lowerCI.rorq = df.boot.pglsl["slope.rorq","lowerCIbca"],
                              upperCI.rorq = df.boot.pglsl["slope.rorq","upperCIbca"],
                              lowerCI.od = df.boot.pglsl["slope.od","lowerCIbca"],
                              upperCI.od = df.boot.pglsl["slope.od","upperCIbca"])
model_param_values <- data.frame(rorqual_slope=a.pglsl[,3],
                                odontocete_slope=a.pglsl[,4])
slope_distributions <- ggplot(model_param_values, aes(fill = fm)) +
  geom_density(aes(odontocete_slope, fill = "#4DBBD5FF", color = NA, alpha = 0.3)) +
  geom_density(aes(rorqual_slope, fill = "#E64B35FF", color = NA, alpha = 0.3)) +
  scale_fill_manual(name = "Feeding mode",
                    values = c("#4DBBD5FF", "#E64B35FF"),
                    labels = c("Single-prey feeders", "Filter feeders")) +
  labs(x = "slope parameter distribution") +
  geom_vline(xintercept = 0, linetype = "dashed", size = 0.8) +
  geom_vline(data=model_param, aes(xintercept=slope.od),
            color = "#4DBBD5FF", linetype=1, size = 0.7) +

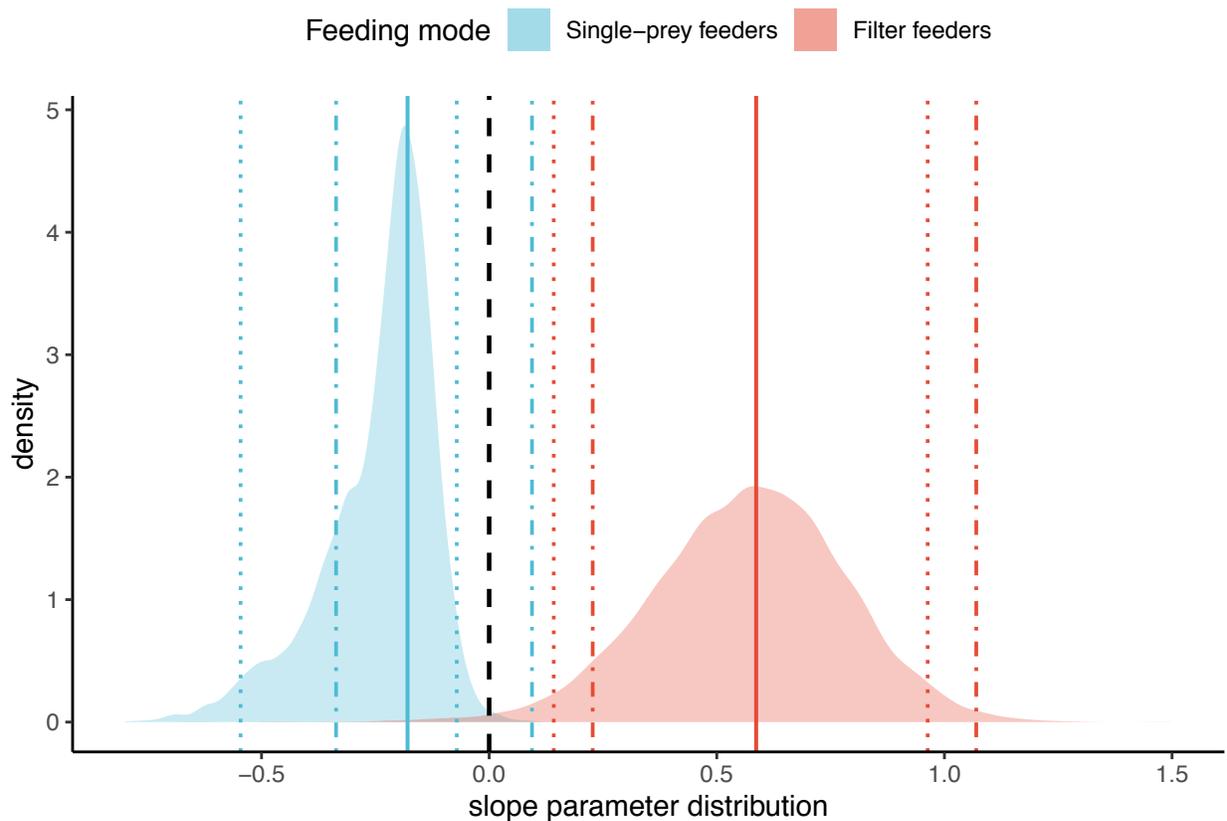
```

```

geom_vline(data=model_param, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=slope.rorq),
  color = "#E64B35FF", linetype=1, size = 0.7) +
geom_vline(data=model_param, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=3, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.rorq),
  color = "#E64B35FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=lowerCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
geom_vline(data=model_param_bca, aes(xintercept=upperCI.od),
  color = "#4DBBD5FF", linetype=4, size = 0.65) +
xlim(-0.8,1.5) +
theme_classic() + theme(legend.position = "top")
slope_distributions

```

Warning: Removed 3 rows containing non-finite values (stat_density).



```

rn <- rownames(df.boot.pgls)
rownames(df.boot.pgls) <- c("intercept filter", "intercept single-prey",

```

```

      "slope filter", "slope single-prey")
knitr::kable(df.boot.pgls,
  caption = "pGLS 95% Bootstrap Pctl CI",
  format = "latex", booktabs = TRUE, escape = FALSE, digits = 4) %>%
  kable_styling(latex_options = "hold_position")

```

Table 22: pGLS 95% Bootstrap Pctl CI

	obs	bootest	lowerCI	upperCI	lowerCIbca	upperCIbca
intercept filter	-1.1283	-1.0683	-2.8599	0.8215	-3.2486	0.5036
intercept single-prey	0.6051	0.8314	0.2514	1.9901	-0.2790	1.2063
slope filter	0.5865	0.5702	0.1419	0.9632	0.2275	1.0699
slope single-prey	-0.1792	-0.2398	-0.5460	-0.0712	-0.3362	0.0941

```
rownames(df.boot.pgls) <- rn
```

8.6 Extract summary statistics

```

specify_decimal <- function(x, k) trimws(format(round(x, k), nsmall = k))

res.df.ols <- m.ols$dim$N - m.ols$dim$p

res.df.pgls <- m.pgls.nlme$dim$N - m.pgls.nlme$dim$p

intercepts.od.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.od", "upperCIbca"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.ols["intercept.od", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.od", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.od", "upperCIbca"], 4),
  ")"))

intercepts.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCI"], 4),
  ")"),
  paste0(specify_decimal(df.boot.pgls["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.pgls["intercept.rorq", "lowerCIbca"], 4),
  " - ", specify_decimal(df.boot.pgls["intercept.rorq", "upperCIbca"],
  4), ")"),
  paste0(specify_decimal(df.boot.ols["intercept.rorq", "obs"], 4),
  " (", specify_decimal(df.boot.ols["intercept.rorq", "lowerCI"], 4),
  " - ", specify_decimal(df.boot.ols["intercept.rorq", "upperCI"], 4),
  ")"))

```

```

paste0(specify_decimal(df.boot.ols["intercept.rorq","obs"],4),
" (", specify_decimal(df.boot.ols["intercept.rorq","lowerCIbca"],4),
" - ", specify_decimal(df.boot.ols["intercept.rorq","upperCIbca"],4),
")"))

slopes.od.ci <- rbind(paste0(specify_decimal(df.boot.pgls["slope.od","obs"],4)," (",
specify_decimal(df.boot.pgls["slope.od","lowerCI"],4)," - ",
specify_decimal(df.boot.pgls["slope.od","upperCI"],4),")"),
paste0(specify_decimal(df.boot.pgls["slope.od","obs"],4)," (",
specify_decimal(df.boot.pgls["slope.od","lowerCIbca"],4)," - ",
specify_decimal(df.boot.pgls["slope.od","upperCIbca"],4),")"),
paste0(specify_decimal(df.boot.ols["slope.od","obs"],4)," (",
specify_decimal(df.boot.ols["slope.od","lowerCI"],4)," - ",
specify_decimal(df.boot.ols["slope.od","upperCI"],4),")"),
paste0(specify_decimal(df.boot.ols["slope.od","obs"],4)," (",
specify_decimal(df.boot.ols["slope.od","lowerCIbca"],4)," - ",
specify_decimal(df.boot.ols["slope.od","upperCIbca"],4),")"))

slopes.rorq.ci <- rbind(paste0(specify_decimal(df.boot.pgls["slope.rorq","obs"],4)," (",
specify_decimal(df.boot.pgls["slope.rorq","lowerCI"],4)," - ",
specify_decimal(df.boot.pgls["slope.rorq","upperCI"],4),")"),
paste0(specify_decimal(df.boot.pgls["slope.rorq","obs"],4)," (",
specify_decimal(df.boot.pgls["slope.rorq","lowerCIbca"],4),
" - ", specify_decimal(df.boot.pgls["slope.rorq","upperCIbca"],
4),")"),
paste0(specify_decimal(df.boot.ols["slope.rorq","obs"],4)," (",
specify_decimal(df.boot.ols["slope.rorq","lowerCI"],4)," - ",
specify_decimal(df.boot.ols["slope.rorq","upperCI"],4),")"),
paste0(specify_decimal(df.boot.ols["slope.rorq","obs"],4)," (",
specify_decimal(df.boot.ols["slope.rorq","lowerCIbca"],4),
" - ", specify_decimal(df.boot.ols["slope.rorq","upperCIbca"],
4),")"))

a.od.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgls["intercept.od","obs"]),4)," (",
specify_decimal(10^(df.boot.pgls["intercept.od","lowerCI"]),4),
" - ", specify_decimal(10^(df.boot.pgls["intercept.od","upperCI"]),4),
")"),
paste0(specify_decimal(10^(df.boot.pgls["intercept.od","obs"]),4)," (",
specify_decimal(10^(df.boot.pgls["intercept.od","lowerCIbca"]),4),
" - ", specify_decimal(10^(df.boot.pgls["intercept.od","upperCIbca"]),4),
")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od","obs"]),4)," (",
specify_decimal(10^(df.boot.ols["intercept.od","lowerCI"]),4)," - ",
specify_decimal(10^(df.boot.ols["intercept.od","upperCI"]),4),")"),
paste0(specify_decimal(10^(df.boot.ols["intercept.od","obs"]),4)," (",
specify_decimal(10^(df.boot.ols["intercept.od","lowerCIbca"]),4),
" - ", specify_decimal(10^(df.boot.ols["intercept.od","upperCIbca"])),
4),")"))

a.rorq.ci <- rbind(paste0(specify_decimal(10^(df.boot.pgls["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.pgls["intercept.rorq","lowerCI"]),5),
" - ", specify_decimal(10^(df.boot.pgls["intercept.rorq","upperCI"]),4),")"),
paste0(specify_decimal(10^(df.boot.pgls["intercept.rorq","obs"]),4),
" (", specify_decimal(10^(df.boot.pgls["intercept.rorq","lowerCIbca"]),5),
" - ", specify_decimal(10^(df.boot.pgls["intercept.rorq","upperCIbca"]),4),

```

Table 23: Model summary statistics

	<i>Filter feeders</i>		<i>Single-prey feeders</i>		RSE	tot.df	res.df
	slope*	intercept	slope	intercept			
pGLS	0.5865 (0.1419 - 0.9632)	-1.1283 (-2.8599 - 0.8215)	-0.1792 (-0.5460 - -0.0712)	0.6051 (0.2514 - 1.9901)	0.3271		
	0.5865 (0.2275 - 1.0699)	-1.1283 (-3.2486 - 0.5036)	-0.1792 (-0.3362 - 0.0941)	0.6051 (-0.2790 - 1.2063)	0.3271		
OLS	0.6493 (0.2156 - 1.0076)	-1.3744 (-3.0244 - 0.5363)	-0.3162 (-0.6875 - -0.2494)	1.1250 (0.9186 - 2.4295)	0.3462	13	9
	0.6493 (0.3169 - 1.1311)	-1.3744 (-3.4994 - 0.1281)	-0.3162 (-0.4868 - -0.1843)	1.1250 (0.6588 - 1.7316)	0.3462		

Note:

* Throughout the table, values in brackets represent 95% confidence intervals: percentile in shaded rows, BCa in non-shaded rows.

```

    ")),
    paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
    " (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCI"]),5),
    " - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCI"]),4),")"),
    paste0(specify_decimal(10^(df.boot.ols["intercept.rorq","obs"]),4),
    " (", specify_decimal(10^(df.boot.ols["intercept.rorq","lowerCIbca"]),5),
    " - ", specify_decimal(10^(df.boot.ols["intercept.rorq","upperCIbca"]),4),
    ")))

RSE <- rbind(specify_decimal(t(t(rep(as.numeric(m.pgl$nlme$sigma),2))),4),
             specify_decimal(t(t(rep(as.numeric(m.ols$sigma),2))),4))
df <- cbind(t(t(c(rep(m.pgl$nlme$dim$N,2), rep(m.ols$dim$N,2)))),
           t(t(c(rep(res.df.pgl$2), rep(res.df.ols,2)))))
models <- rbind(t(t(rep("pGLS",2))),t(t(rep("OLS",2))))

outputs <- cbind(models, slopes.rorq.ci, intercepts.rorq.ci, slopes.od.ci,
                 intercepts.od.ci, RSE, df)
df.outputs <- data.frame(outputs, check.rows = TRUE, check.names = TRUE)
names(df.outputs) <- c("","slope","intercept","slope","intercept","RSE", "tot.df","res.df")
names(df.outputs)[2] <- paste0(names(df.outputs)[2],
                              footnote_marker_symbol(1))

knitr::kable(df.outputs,
             caption = "Model summary statistics",
             format = "latex", booktabs = TRUE, escape = FALSE) %>%
  kable_styling(latex_options = "scale_down") %>%
  row_spec(0, bold = T) %>%
  row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
  column_spec(c(1,(ncol(df.outputs)-1):ncol(df.outputs))-1,
             background = "white") %>%
  column_spec(1, bold = T) %>%
  collapse_rows(columns = c(1,(ncol(df.outputs)-1):ncol(df.outputs))) %>%
  add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2,
                    " " = 3), bold = T, italic = T) %>%
  footnote(general = "", general_title = "Note:",
          symbol = paste0("Throughout the table, values in brackets",
                          " represent 95% confidence intervals: ",
                          "percentile in shaded rows, BCa in non-shaded rows."),
          symbol_title = "", title_format = "italic",
          footnote_as_chunk = T)

alloout <- cbind(models, a.rorq.ci, slopes.rorq.ci, a.od.ci, slopes.od.ci)
df.allo <- data.frame(alloout, check.rows = TRUE, check.names = TRUE)
names(df.allo) <- c("","a","b","a","b")

```

Table 24: Transformed to allometric equations

	<i>Filter feeders</i>		<i>Single-prey feeders</i>	
	a*	b	a	b
pGLS	0.0744 (0.00138 - 6.6298)	0.5865 (0.1419 - 0.9632)	4.0278 (1.7841 - 97.7397)	-0.1792 (-0.5460 - -0.0712)
	0.0744 (0.00056 - 3.1883)	0.5865 (0.2275 - 1.0699)	4.0278 (0.5260 - 16.0791)	-0.1792 (-0.3362 - 0.0941)
OLS	0.0422 (0.00095 - 3.4382)	0.6493 (0.2156 - 1.0076)	13.3350 (8.2910 - 268.8233)	-0.3162 (-0.6875 - -0.2494)
	0.0422 (0.00032 - 1.3430)	0.6493 (0.3169 - 1.1311)	13.3350 (4.5578 - 53.9016)	-0.3162 (-0.4868 - -0.1843)

* Throughout the table, values in brackets represent 95% confidence intervals.: percentile in shaded rows, BCa in non-shaded rows.

```
names(df.allo)[2] <- paste0(names(df.allo)[2], footnote_marker_symbol(1))
knitr::kable(df.allo,
  caption = "Transformed to allometric equations",
  format = "latex", booktabs = TRUE, escape = FALSE) %>%
  kable_styling(latex_options = "scale_down") %>%
  row_spec(0, bold = T) %>%
  row_spec(c(1,3)-1, extra_latex_after = "\\rowcolor{gray!6}") %>%
  column_spec(1, bold = T) %>%
  collapse_rows(columns = 1) %>%
  add_header_above(c(" " = 1, "Filter feeders" = 2, "Single-prey feeders" = 2),
    bold = T, italic = T) %>%
  footnote(symbol = paste0("Throughout the table, values in brackets",
    " represent 95% confidence intervals.: ",
    "percentile in shaded rows, BCa in non-shaded rows."),
    symbol_title = "", threeparttable = TRUE, footnote_as_chunk = T)
```

8.7 Plot best models (OLS - dashed, PGLS - solid)

```
pgls.fit <- predict(m.pgls.nlme)
ols.fit <- predict(m.ols)
predframe <- with(smydata, data.frame(species, Group, x_mean, y_mean = pgls.fit))
predframe2 <- with(smydata, data.frame(species, Group, x_mean, y_mean = ols.fit))

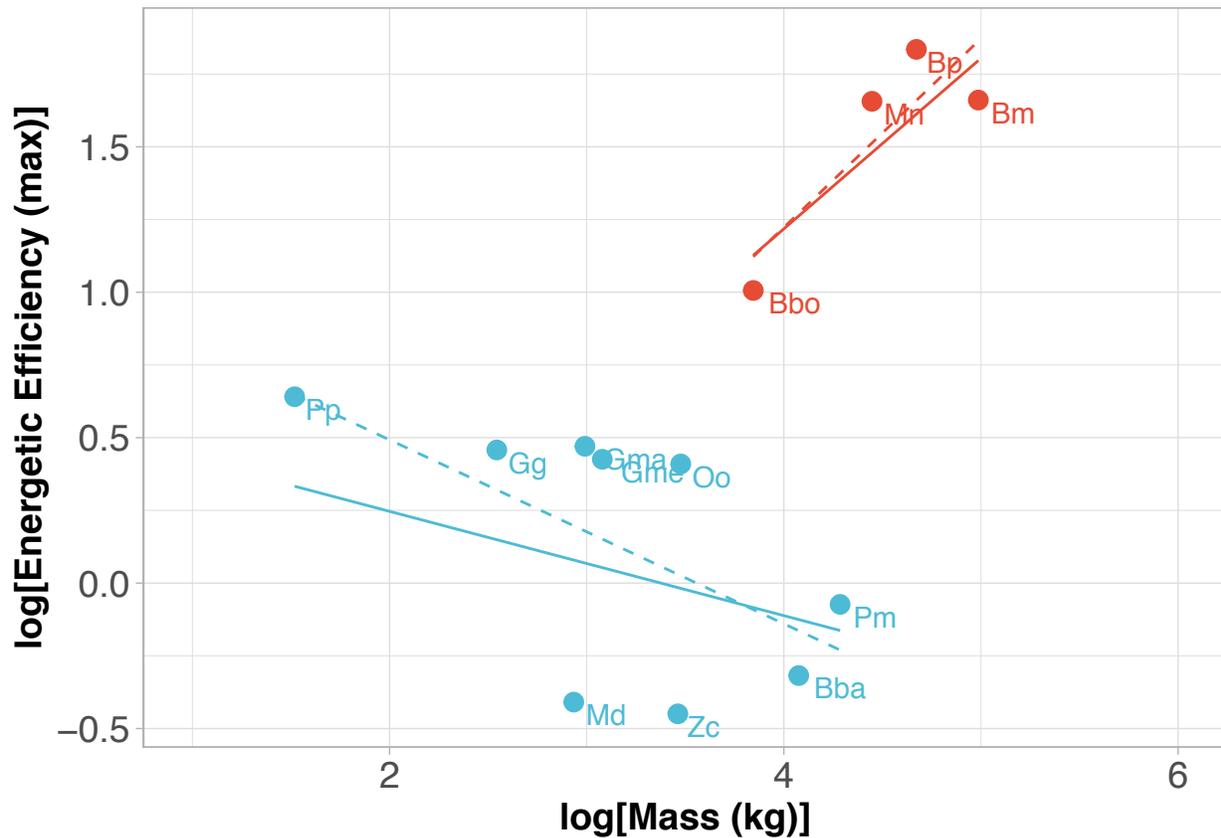
fig_4.75 <- ggplot(smydata, aes(x_mean, y_mean, color = Group)) +
  geom_point(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
    size = 3) +
  geom_point(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
    size = 3) +
  geom_line(data = dplyr::filter(predframe, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 1) +
  geom_line(data = dplyr::filter(predframe2, Group == "Rorqual"), color = "#E64B35FF",
    linetype = 2) +
  geom_line(data = dplyr::filter(predframe2, Group == "Odontocete"), color = "#4DBBD5FF",
    linetype = 2) +
  guides(size = FALSE, color = FALSE) +
  theme_light() + theme(legend.position = "top") +
  xlim(1,6) +
  theme(axis.text = element_text(size = 14), axis.title = element_text(size = 14,
```

```

labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]" +
      face = "bold")) +
geom_text(data = dplyr::filter(smydata, Group == "Rorqual"), color = "#E64B35FF",
          aes(label = abbreviation, hjust = -0.3, vjust = 1.1) +
geom_text(data = dplyr::filter(smydata, Group == "Odontocete"), color = "#4DBBD5FF",
          aes(label = abbreviation), hjust = -0.3, vjust = 1.1)

```

fig_4.75



8.7.1 Construct output table

```

df.out <- smydata[,c("species", "fm", "x_mean", "y_mean")]
df.out$fitted_ols <- fitted(m.ols)
df.out$fitted_pgls <- fitted(m.pgls.nlme)
rownames(df.out) <- NULL
kable(df.out,
      caption = "Model outputs",
      format = "latex", booktabs = TRUE, digits = 4) %>%
kable_styling(latex_options = "scale_down")

```

8.8 Quick clean up

```

m.75.pgls.nlme <- m.pgls.nlme
df.75.outputs <- df.outputs
m.75.ols <- m.ols

```

Table 25: Model outputs

species	fm	x_mean	y_mean	fitted_ols	fitted_pglis
Balaenoptera_bonaerensis	Filter	3.8451	1.0061	1.1221	1.1269
Balaenoptera_musculus	Filter	4.9868	1.6606	1.8634	1.7965
Balaenoptera_physalus	Filter	4.6725	1.8348	1.6593	1.6122
Berardius_bairdii	Single-prey	4.0755	-0.3180	-0.1638	-0.1254
Globicephala_macrorhynchus	Single-prey	2.9912	0.4703	0.1791	0.0689
Globicephala_melas	Single-prey	3.0792	0.4257	0.1513	0.0532
Grampus_griseus	Single-prey	2.5441	0.4577	0.3205	0.1491
Megaptera_novaeangliae	Filter	4.4472	1.6563	1.5130	1.4800
Mesoplodon_densirostris	Single-prey	2.9345	-0.4092	0.1970	0.0791
Orcinus_orca	Single-prey	3.4771	0.4096	0.0254	-0.0182
Phocoena_phocoena	Single-prey	1.5185	0.6407	0.6448	0.3329
Physeter_macrocephalus	Single-prey	4.2856	-0.0732	-0.2302	-0.1631
Ziphius_cavirostris	Single-prey	3.4624	-0.4496	0.0301	-0.0155

```
to.keep <- c(to.keep, "m.75.pglis.nlme", "df.75.outputs", "smydata", "m.75.ols")
rm(list=setdiff(ls(), to.keep))
```

9 Combine best models

9.1 Combine parameters into a single data frame

```
m.45.ols.param <- as.data.frame(t(summary(m.45.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.45.ols.param <- m.45.ols.param[5:8]
m.61.ols.param <- as.data.frame(t(summary(m.61.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.61.ols.param <- m.61.ols.param[5:8]
m.68.ols.param <- as.data.frame(t(summary(m.68.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.68.ols.param <- m.68.ols.param[5:8]
m.75.ols.param <- as.data.frame(t(summary(m.75.ols)$tTable[,1])) %>%
  mutate(intercept.rorq = `(Intercept)`,
         intercept.od = `(Intercept)`+`fmSingle-prey`,
         slope.rorq = `x_mean`, slope.od = `x_mean`+`fmSingle-prey:x_mean`)
m.75.ols.param <- m.75.ols.param[5:8]

m.ols.param.all <- rbind(m.45.ols.param,m.61.ols.param,m.68.ols.param,m.75.ols.param)
rownames(m.ols.param.all) <- c(".45", ".61", ".68", ".75")
fil <- paste0("C:\\Users\\Danuta\\Dropbox\\foragescaling\\pglis\\",
```

```

"Figure4_m_ols_param_all.rds")
saveRDS(m.ols.param.all,fil)

```

9.2 Plot all models (OLS - thin lines, pGLS - thick lines)

Metabolic rate scaling coefficients: 0.45 - solid line; 0.61 - dashed line; 0.68 - dot-dash line; 0.75 - dotted line.

```

m.45.fit <- predict(m.45.pglsl.nlme)
m.61.fit <- predict(m.61.pglsl.nlme)
m.68.fit <- predict(m.68.pglsl.nlme)
m.75.fit <- predict(m.75.pglsl.nlme)

predframe.all <- with(smydata, data.frame(species, Group, x_mean, y45 = m.45.fit,
                                         y61 = m.61.fit, y68 = m.68.fit, y75 = m.75.fit))

m.45.olsfit <- predict(m.45.ols)
m.61.olsfit <- predict(m.61.ols)
m.68.olsfit <- predict(m.68.ols)
m.75.olsfit <- predict(m.75.ols)

predframe.all.ols <- with(smydata, data.frame(species, Group, x_mean, y45 = m.45.olsfit,
                                             y61 = m.61.olsfit, y68 = m.68.olsfit,
                                             y75 = m.75.olsfit))

fig_4_fin <- ggplot(smydata, aes(x_mean, y = value, color = Group)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Rorqual"),
           color = "#E64B35FF", size = 1, aes(y = y45)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Odontocete"),
           color = "#4DBBD5FF", size = 1, aes(y = y45)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Rorqual"),
           color = "#E64B35FF", linetype = 2, size = 1, aes(y = y61)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Odontocete"),
           color = "#4DBBD5FF", linetype = 2, size = 1, aes(y = y61)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Rorqual"),
           color = "#E64B35FF", linetype = 4, size = 1, aes(y = y68)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Odontocete"),
           color = "#4DBBD5FF", linetype = 4, size = 1, aes(y = y68)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Rorqual"),
           color = "#E64B35FF", linetype = 3, size = 1, aes(y = y75)) +
  geom_line(data = dplyr::filter(predframe.all, Group == "Odontocete"),
           color = "#4DBBD5FF", linetype = 3, size = 1, aes(y = y75)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Rorqual"),
           color = "#E64B35FF", size = 0.5, aes(y = y45)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Odontocete"),
           color = "#4DBBD5FF", size = 0.5, aes(y = y45)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Rorqual"),
           color = "#E64B35FF", linetype = 2, size = 0.5, aes(y = y61)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Odontocete"),
           color = "#4DBBD5FF", linetype = 2, size = 0.5, aes(y = y61)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Rorqual"),
           color = "#E64B35FF", linetype = 4, size = 0.5, aes(y = y68)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Odontocete"),
           color = "#4DBBD5FF", linetype = 4, size = 0.5, aes(y = y68)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Rorqual"),
           color = "#E64B35FF", linetype = 3, size = 0.5, aes(y = y75)) +
  geom_line(data = dplyr::filter(predframe.all.ols, Group == "Odontocete"),
           color = "#4DBBD5FF", linetype = 3, size = 0.5, aes(y = y75))

```

```

    color = "#E64B35FF", linetype = 3, size = 0.5, aes(y = y75)) +
geom_line(data = dplyr::filter(predframe.all.ols, Group == "Odontocete"),
    color = "#4DBBD5FF", linetype = 3, size = 0.5, aes(y = y75)) +
theme_light() + theme(legend.position = "top") +
xlim(1,6) +
guides(size = FALSE, color = FALSE) +
theme(axis.text = element_text(size = 14), axis.title = element_text(size = 16,
    face = "bold")) +
labs(x = "log[Mass (kg)]", y = "log[Energetic Efficiency (max)]")

```

fig_4_fin

