Supplemental Analysis

ANONYMIZED

Phylogenetic Specialism

Here we quantify the phylogenetic specialism of frugivores. To do this, we'll employ the `ses.mpd()` from the `picante` library. This function allows us to use a null model of our choice to find the standard effect size of pairwise distance in communities. In our case, we're interested in pairwise mean phylogenetic distance, and our "communities" are the total sets of plants that each frugivore interact with. We employ an independent swap null model, where the tip labels of the underlying phylogenetic tree are effectively swapped a given number of times (Gotelli, 2000).

```
set.seed(1)
load("../Data.nosync/DataSources/BIEN_subtree.Rda")
# Create cophenetic distance matrix
dmatrix <- stats::cophenetic(BIEN_subtree)
# Standardize by the maximum value
dmatrix <- dmatrix/max(dmatrix)

# String manipulation so tree tip names match
birds$plant_Tips <- gsub(pattern = " ", replace = "_", x = birds$Plant_Species)
# Only include plant species for which phylogenetic info is
# available
birds_phylo <- dplyr::filter(birds, plant_Tips %in% BIEN_subtree$tip.label)

comm <- as.data.frame.matrix(table(birds_phylo$Frugivore_Species,
    birds_phylo$plant_Tips))  #Binary, unweighted intraction matrix.
# The main function. Randomly swap tip labels 999 times and
# see whether the observed sets of plant interactors for
# each species is more phylogenetically clustered than this
# assumed distribution
test_mpd <- picante::ses.mpd(comm, dmatrix, null.model = "independentswap",
    abundance.weighted = FALSE, runs = 999, iterations = 1000)
# Remove singleton species (can't compute a phylogenetic
# distance)
phylospec <- dplyr::filter(test_mpd, ntaxa > 1)
# Remove singleton species (can't compute a phylogenetic
# distance)
sig <- nrow(dplyr::filter(phylospec, mpd.obs.p < 0.05))
print(sig)
```

```
## [1] 9
```

We see that the number of frugivores whose plant partners are more significantly related than expected compared to a random draw is equal to 9 (The exact value may change slightly when the seed is changed due to the stochastic nature of the model, but the results should be qualitatively similar each time)

Calculating the number of unique species in the dataset

In our dataset, we observe interactions between 242 unique frugivore species and 458 unique plant species.

Degree Distributions of Mutualistic Partners

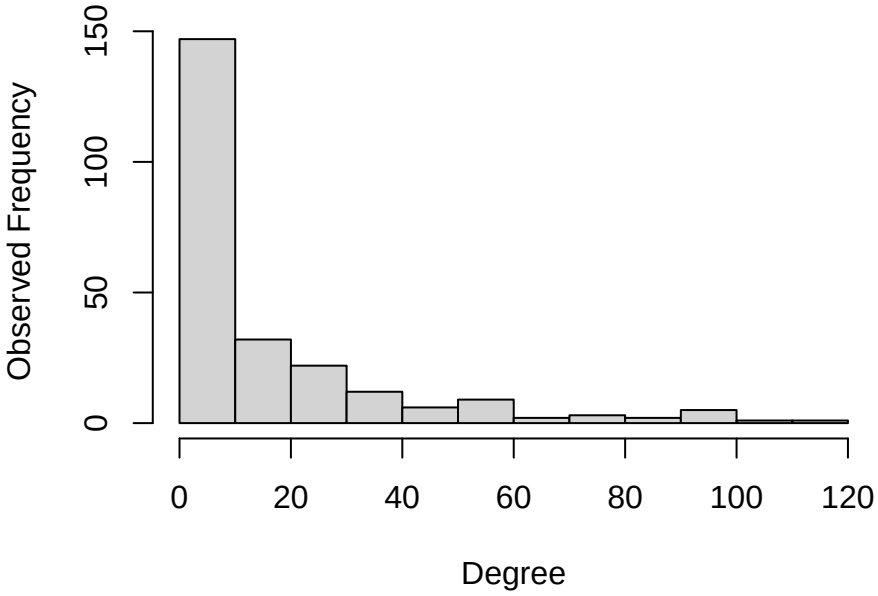

Figure S1: Observed degree distribution of avian frugivore species included in our analyses. X-axis represents node degree, or number of unique partners. Degree for frugivores ranged from 1 (53 species) to 120 unique plant interactions recorded for *Turdus rufiventris*; median frugivore degree was 5.

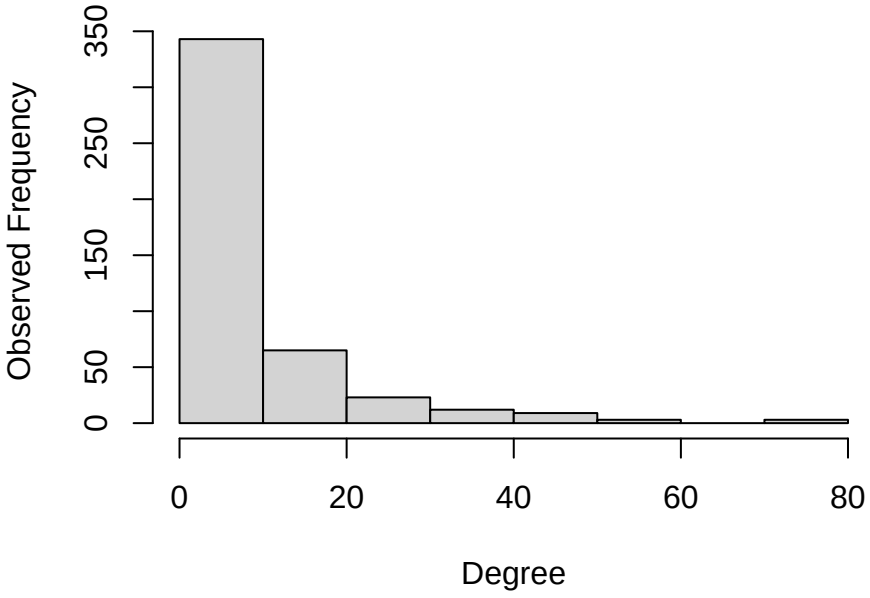

Figure S2: Observed degree distribution of plant species included in our analyses. X-axis represents node degree, or number of unique partners. Degree on average tended to be lower for plants than frugivores; plant node degree ranged from 1 (128 species) to 80 unique frugivorous interactions recorded for *Myrsine coriacea*; median plant degree was 5.

```
## [1] 37.5
```

Table S1: Number of interactions, unique bird species, and unique plant species used in each model.

Incomplete modeling cases were dropped from each model; the remaining number of complete cases was shown below.

```
kable(counTable, col.names = c("Model", "Number of Links", "Number of Bird species",
    "Number of Plant spp"))
```

Model	Number of Links	Number of Bird species	Number of Plant spp
Latent	3856	242	458
Phy	3276	225	377
Traits	2813	159	283
PhyLatent	3276	225	377
PhyTraits	2492	157	246
TraitsLatent	2813	159	283
Trio	2492	157	246

Impact of Phylogenetic Imputation

In our main text results, for the 46 plant species and 2 bird species which did not appear in the appropriate phylogenies but did have congenerics, we randomly assigned those species to be a polytomy within their parent genera. When these species are excluded from the analysis, our results are still qualitatively indistinguishable. Below are versions of the three main-text figures with those 48 species removed from the network.

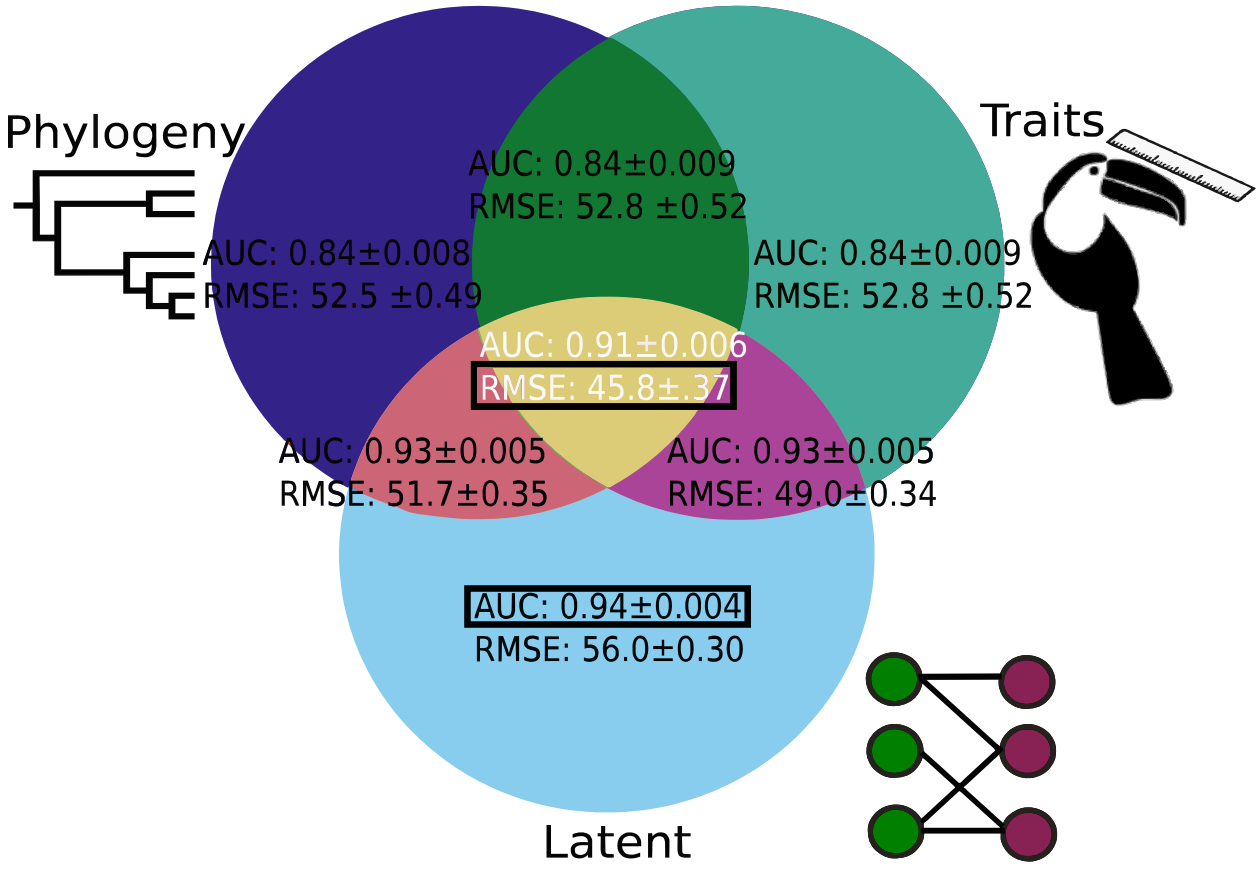

Figure S3: Summary performance metrics of all 7 models *when species without full phylogenetic information are excluded*, as measured by area under the receiver operating characteristic curve (AUC) and root mean squared error (RMSE); highest performing models for each metric are outlined in black. Mean metric values are presented from 100 replicates of each model structure alongside standard deviation. As in the main text, model discriminatory power between links and non-links is maximized by including latent structural features, with the inclusion of trait, phylogenetic information, or both actually slightly decreasing discriminatory power.

However, inclusion of trait and phylogenetic information, while not improving AUC, does increase overall model accuracy as measured by mean root squared error.

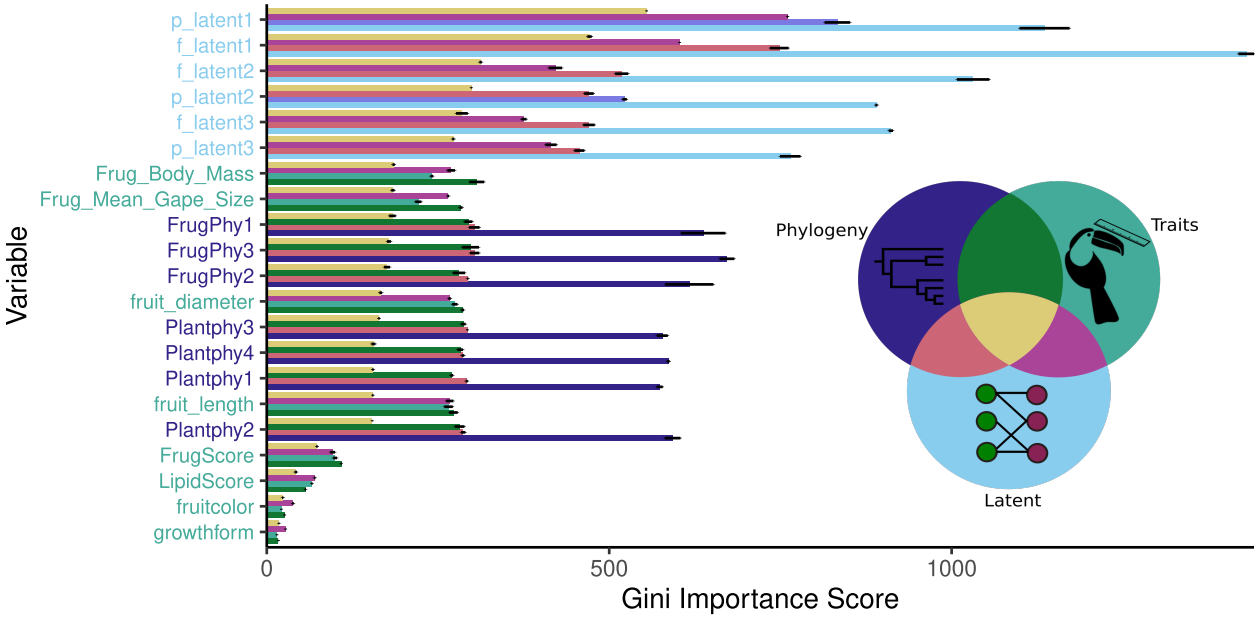

Figure S4: Variable importance across all models as measure by Gini importance score *when species without full phylogenetic information are excluded*; color scheme is consistent with figure S3. As in the maintext,latent traits were consistently the most important variables for prediction. These were followed by continuous frugivore traits (body mass, gape size), and frugivore phylogenetic axes. Plant phylogenies and continuous trait information were generally less important for prediction than frugivore traits. Categorical plant traits (Lipid content, fruit color, growth form) were the least important variables for prediction.

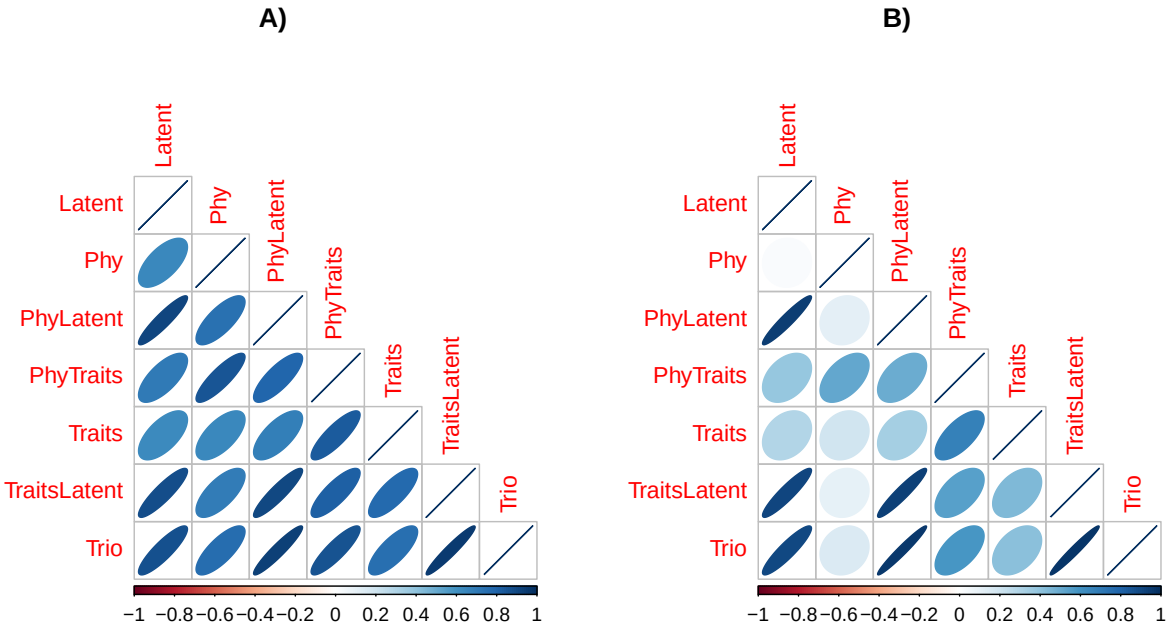

Figure S4: Pairwise Spearman's rank correlations of link suitabilities across models for all potential interaction (A), as well as only unobserved interactions (B), *after species without full phylogenetic information*

are excluded. The latter set of links represents both true forbidden links, as well as other potential interactions not observed in our data-set.

Sensitivity to Class Imbalances

In order to deal with the high degree of sparceness in our network, we trim the training set to enforce class balancing. In the maintext, we present the results of a 3:1 ratio unobserved:observed links. Here, show the methods is relatively insensitive to this exact proportion by repeating our analysis with alternative training ratios of 1:1 and 1:10.

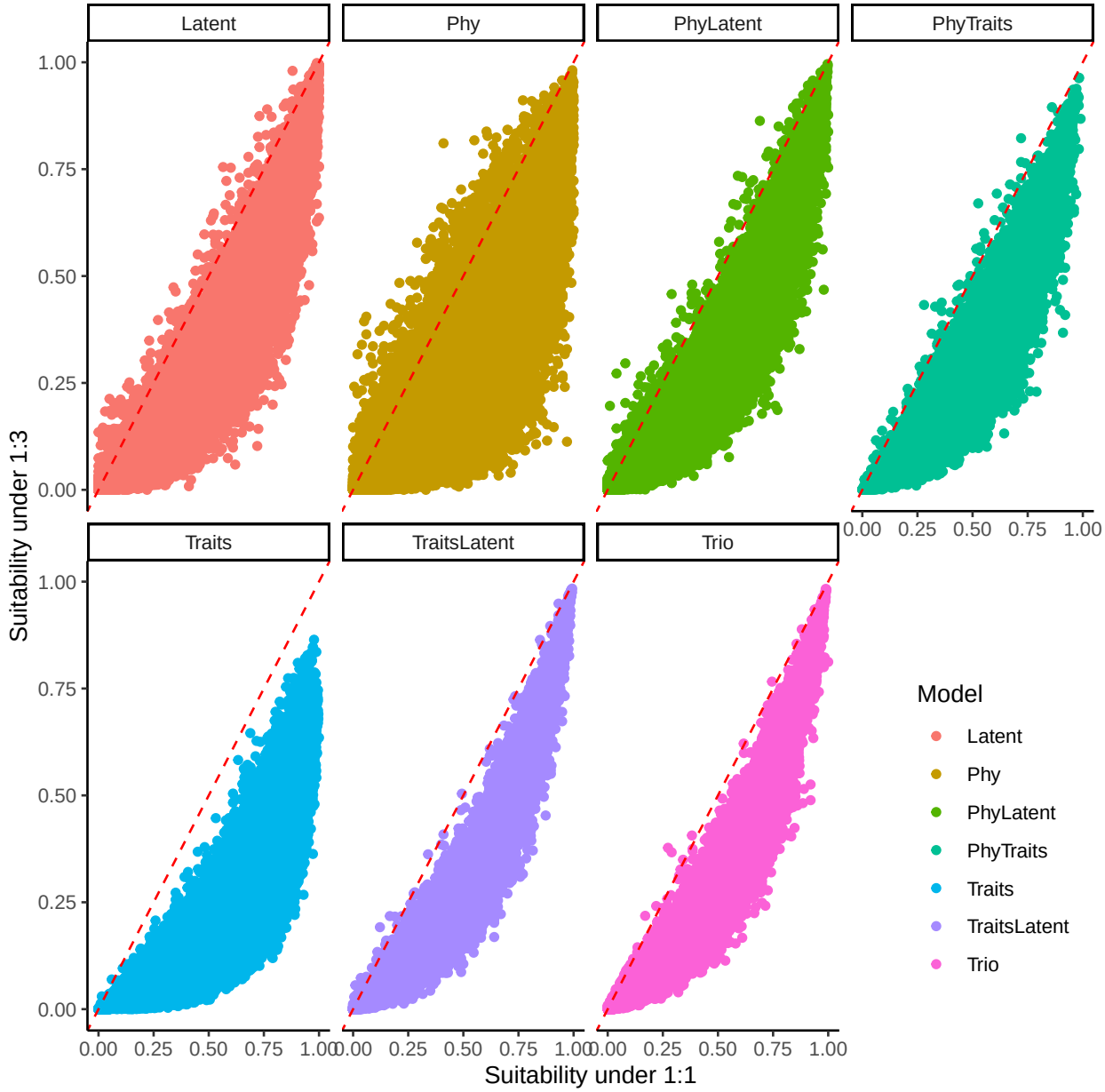

Figure S5: Scatterplot of relative suitability values of the same models trained under either a 1:3 ratio of present to absent interactions (y-axis) or a 1:1 ratio (x-axis); red dashed lines represent a 1:1 line. While the absolute value of suitability values tend to be reduced when the prevalence of true positives is reduced (most points fall below the 1:1 line), the relative rank of suitability values tends is very closely aligned (Spearman's $\rho = 0.964$ across all predictions).

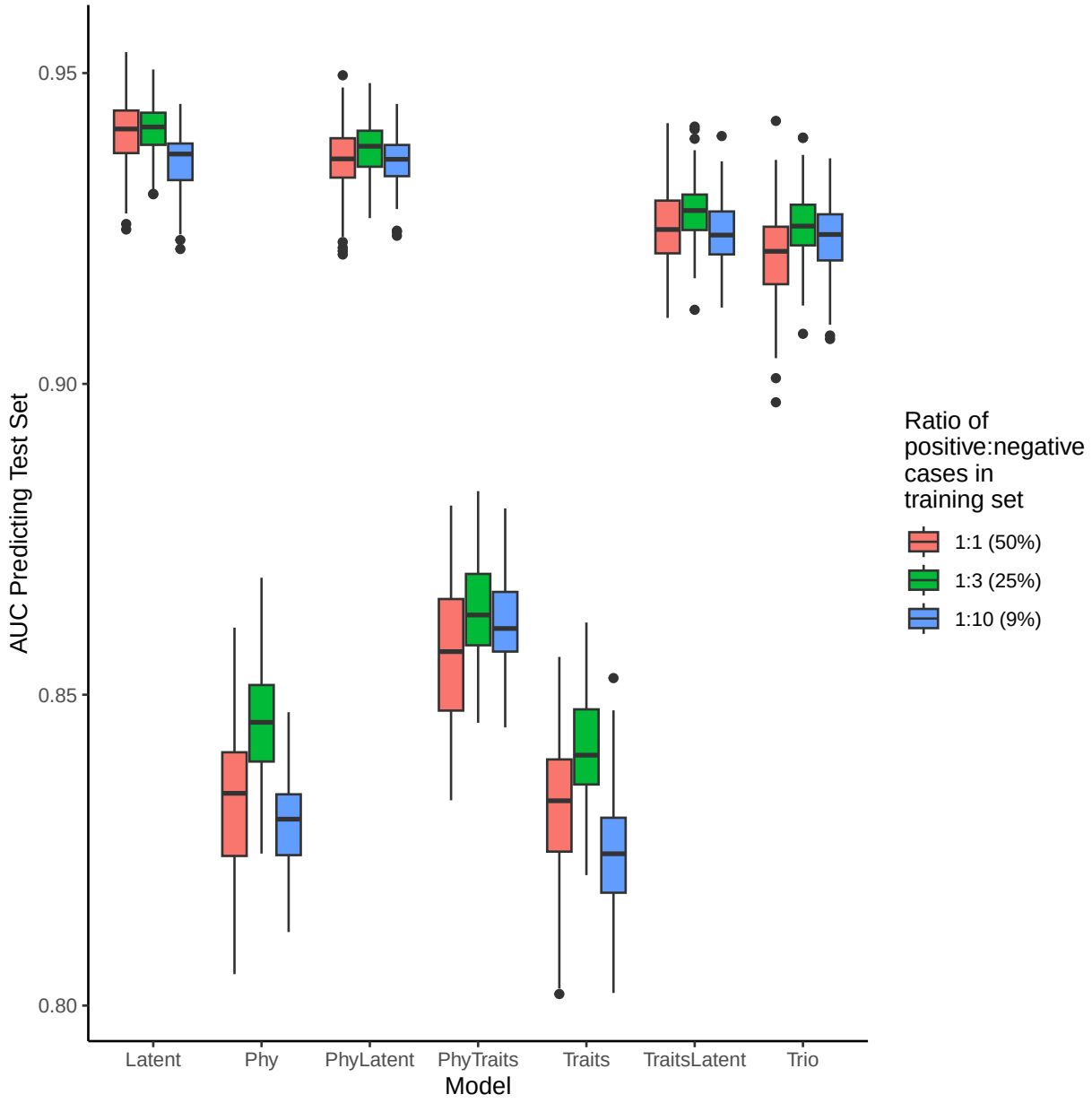

Figure S6: Box plots of AUC values for model predictions on a 20% test set after being conditioned on training data with either 1:1, 1:3, or 1:10 ratios of presence to absence values. We see that while there is some variation in model performance according to training prevalence, the relative performance of each model is still qualitatively the same across the range of prevalence values.

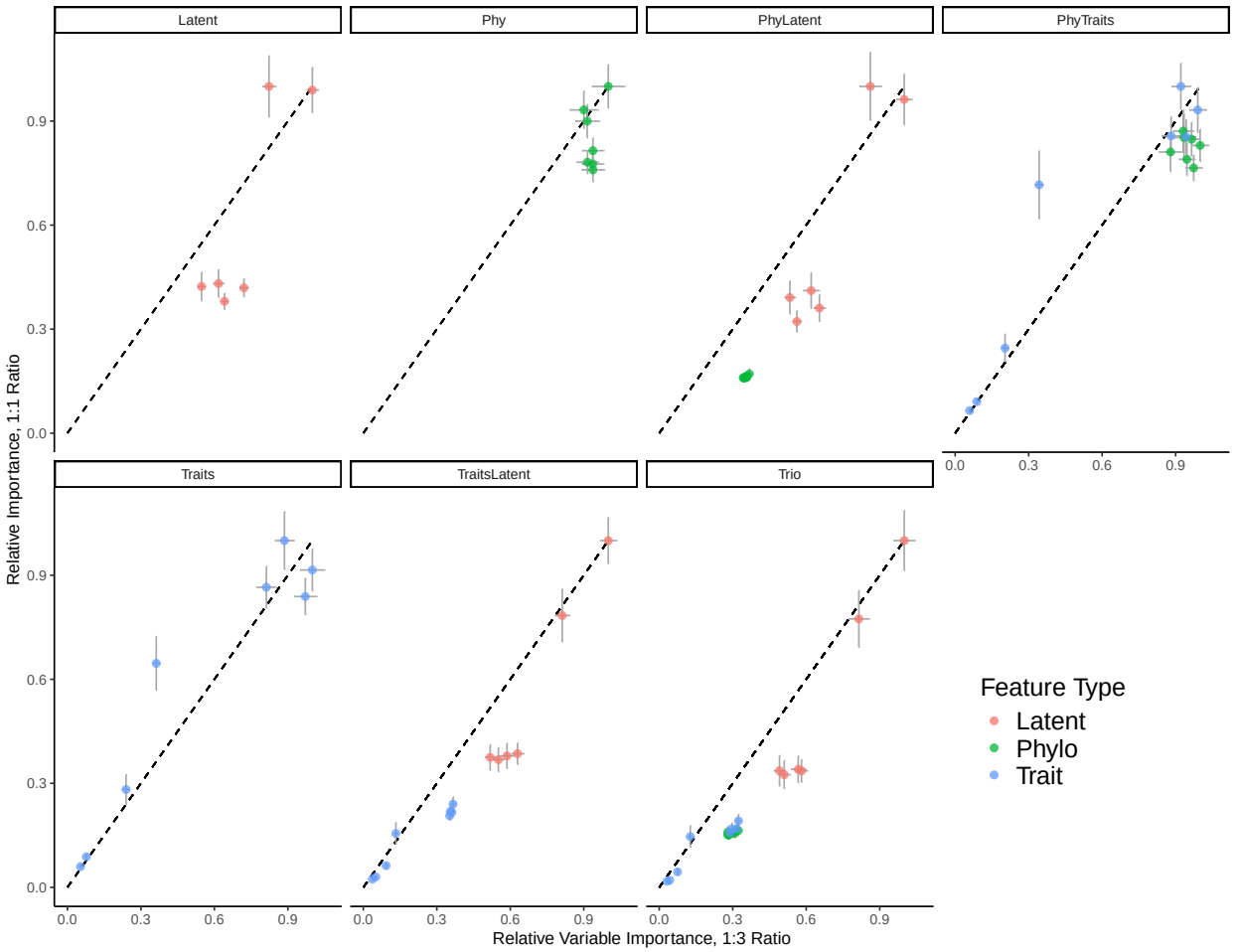

Figure S7: Variable importance between the same models trained on either a 1:1 ratio of present to absent interactions (y-axis) or a 1:3 ratio (x-axis). Dotted black lines represent a 1:1 line; points that fall on the line indicate that relative importance of that variable was unaffected by training prevalence. We see that the marjority of features fall very close to the 1:1 line, including the most important variables (top right of each inset plot).This indicates are models are realtively insensitive to training prevalence.

In Defense of Latent Traits

Latent traits are derived from the observed network, which is an incomplete sample of some underlying "true" network that we're trying to predict. We might expect the ability of latent features to effectively predict links to be different across networks with different sample completeness. Here, we, attempt to characterize that sensitiviy through a null model approach.

The function below takes an arbitray number of plants, frugivores, and "true" links between them and simulates a network based on those characteristics. Each frugivore and plant species is assigned an abundance value randomly drawn from a lognormal distribution. Links are randomly assigned across all potential interactions according to the abundance of both potential partners. Links between abundant species are more likely than links between rare species. While a simplistic, neutral process, this graph generation feature recapitulates long-tailed degree distributions and asymmetric specialization patterns frequently observed in empirical networks.

Given this "true" observed network, we perform single-value decomposition to create latent features, extracting the first three axes of variation for all nodes as in the main text. From these features we create a euclidean distance matrix that represents the pairwise distances of each intraguild node combination in the three-

dimensional space defined by these three latent trait axes. The result is a `f\timesf` and a `p\timesp` distance matrix, where `f` and `p` are the number of frugivores and plants, respectively.

We then degrade this network, removing 50 links from the full set, again sampling non-randomly so that less abundant combinations of species are more likely to go unsampled. This serves to approximate an incomplete sampling process in an empirical system, where links between abundant partners are likely to be observed, but links between rare partners are more likely to be missed. We then repeat the process of single value decomposition and creation of an intriguild pairwise distance matrix. We then use spearman's rank correlations to compare the "true" distance matrix with that produced from the degraded network. If latent features are able to separate the ranks of most nodes even despite degradation, spearman's rank correlation should be high. However, if degrading the network severly alters node clustering, spearman's correlatio should be low between the true and observed distance matrices. By repeating this process across a range of degradation values (each time comparing back to the "true" network), we can start to characterize the sensitivity of latent features to incomplete sampling.

Below, we paramaterize the "true" model with 787 plant species, 242 frugivores, and 3643 "true" links, the same size as our empirically observed network. We repeat the process of degrading the network 50 times, and present average decreases in spearman's rank correlation.

```
# n_plants <- 787; n_frugivores=242; n_links = 3643;
# n_observed = 300; backgroundratio=3 #for debugging

distcordecay <- function(n_plants = 100, n_frugivores = 50, n_links = 500,
    aundweightsamp = T) {
    # Generate all combinations
    interactions <- expand.grid(FrugivoreID = 1:n_frugivores,
        PlantID = 1:n_plants) %>%
        left_join(data.frame(FrugivoreID = 1:n_frugivores, FrugAbund = rlnorm(n_frugivores)),
            by = "FrugivoreID") %>%
        left_join(data.frame(PlantID = 1:n_plants, PlantAbund = rlnorm(n_plants)),
            by = "PlantID") %>%
        mutate(AbundProd = PlantAbund * FrugAbund)  #Make a column of abundance products
    # Assign 'real' interactions using abundance-weighted
    # probabilities
    interactions$real <- 0
    interactions$real[sample(1:nrow(interactions), size = n_links,
        replace = FALSE, prob = interactions$AbundProd)] <- 1

    ############################################################

    # Rename species
    interactions$FrugivoreID <- paste0("F", interactions$FrugivoreID)
    interactions$PlantID <- paste0("P", interactions$PlantID)

    # Perform SVD latent feature reduction
    mat_assym <- xtabs(real ~ FrugivoreID + PlantID, data = interactions)  #Make adj matrix
    decomp <- svd(mat_assym)
    u <- decomp$u
    v <- data.frame(decomp$v)

    # Trait tables
    plantsSVD <- data.frame(PlantID = colnames(mat_assym), Psvd1 = v[,
        1], Psvd2 = v[, 2], Psvd3 = v[, 3])
    frugSVD <- data.frame(FrugivoreID = rownames(mat_assym),
        Fsvd1 = u[, 1], Fsvd2 = u[, 2], Fsvd3 = u[, 3])
```

```
    # Create our true distance matrices to compare to
    pdistrue <- dist(plantsSVD[, 2:4], diag = T, upper = T, method = "euclidean")
    fdistrue <- dist(frugSVD[, 2:4], diag = T, upper = T, method = "euclidean")

    ##############################################################

    output <- NULL
    # Now start subsampling to degrade the network
    for (loss in seq(50, n_links, by = 50)) {
        temp <- interactions
        temp$obs <- 0
        temp$obs[sample(which(interactions$real == 1), size = n_links -
            loss, replace = FALSE, prob = interactions$AbundProd[which(interactions$real ==
            1)])] <- 1

        mat_assym <- xtabs(obs ~ FrugivoreID + PlantID, data = temp)  #Make adj matrix of observed subs
        decomp <- svd(mat_assym)  #perform decomposition
        u <- decomp$u
        v <- data.frame(decomp$v)

        plantsSVD <- data.frame(PlantID = colnames(mat_assym),
            Psvd1 = v[, 1], Psvd2 = v[, 2], Psvd3 = v[, 3])
        frugSVD <- data.frame(FrugivoreID = rownames(mat_assym),
            Fsvd1 = u[, 1], Fsvd2 = u[, 2], Fsvd3 = u[, 3])

        pdistemp <- dist(plantsSVD[, 2:4], diag = T, upper = T,
            method = "euclidean")  #Create distance matrices
        fdistemp <- dist(frugSVD[, 2:4], diag = T, upper = T,
            method = "euclidean")
        pcor <- cor(pdistrue, pdistemp, method = "spearman")  #correlate our real distance matrices wit
        fcor <- cor(fdistrue, fdistemp, method = "spearman")

        ret <- data.frame(loss = loss, plantcor = pcor, frugcor = fcor)  #grab our correlations
        output <- rbind(ret, output)
        # print(paste(loss, 'complete'))
    }
    return(output)
}
```

```
set.seed(1)
decayfull <- NULL
for (i in 1:50) {
    temp <- distcordecay(n_plants = 787, n_frugivores = 242,
        n_links = 3643, aundweightsamp = T)
    temp$run <- i
    decayfull <- rbind(temp, decayfull)
    print(paste(i, "complete"))
}
# save(decayfull, file='decaytestFull.RDA')
```

```
## [1] 0.6450727
```

```
## [1] 0.7685973
```

```
## pdf
```

```
##     2
```

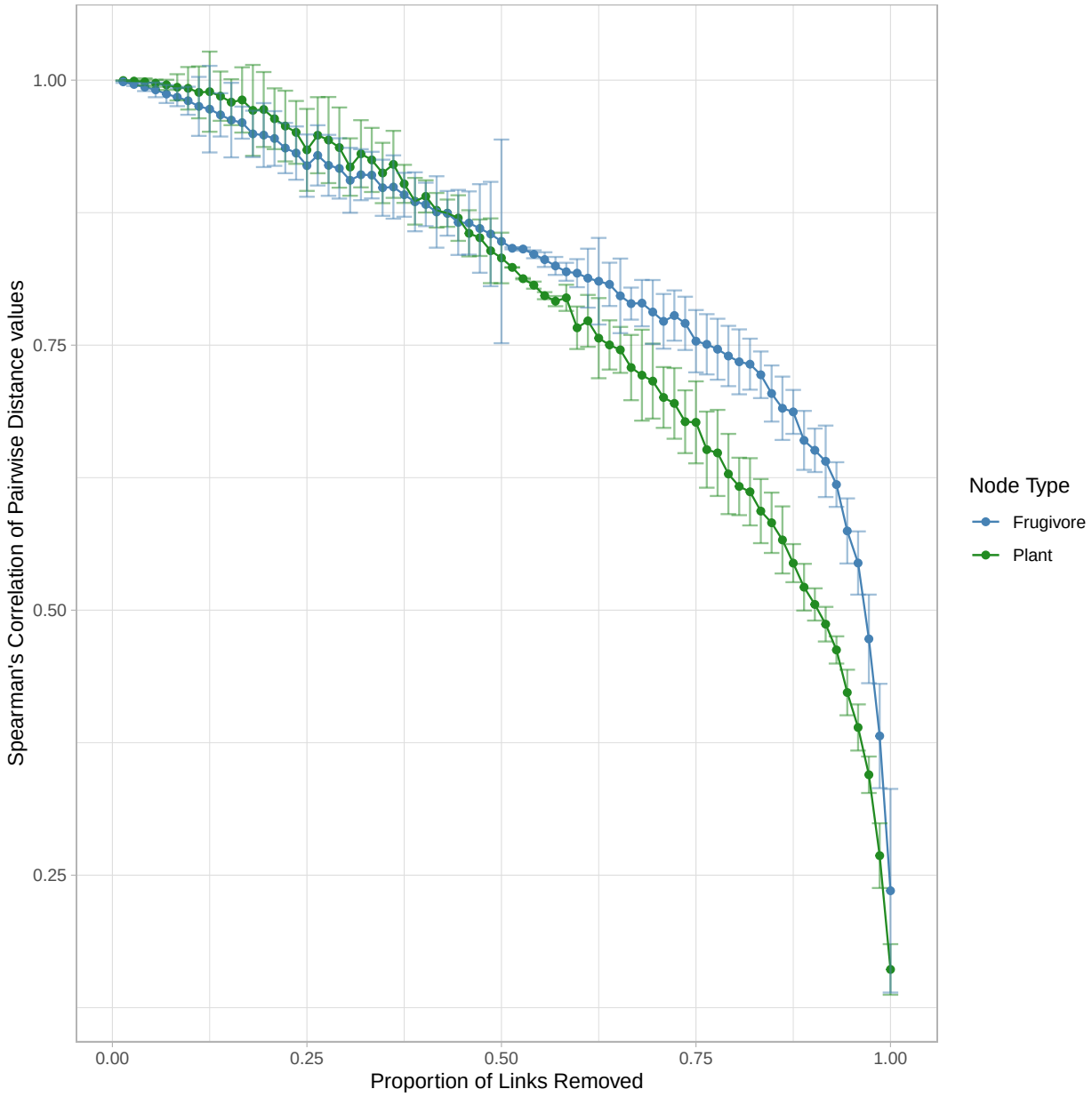

Figure S8: Plot of intraguild latent pairwise distance matrices derived from compared to the true network (spearman's rank correlation); lines represent mean values across 50 iterations, while error bars represent standard deviation. Correlation decreases as links are removed for both frugivore (blue) and plant (green) nodes, but generally decreases slowly until a high proportion of links are removed. Distance matrices exhibit $\rho > 0.75$ for plant and frugivore species even after removing 64.5% and 76.9% of links, respectively.

We also are interested in the ability of latent features to cpature neutral processes given the aforementioned feature of incomplete sampling. Below, we simulate a number of "true" networks where attatchment is again only a function of species abundances drawn from a lognormal distribution.

Below, we load in our analysis function from the main text.

```
# First load in our analysis function from the maintext
woodedWalk_NoSplit <- function(dat, FrugTraits, PlantTraits,
```

```
    class_balancing = FALSE, balance_ratio = 3, output_type = "rfobject") {
    if ((output_type %in% c("rfobject", "predictions")) == FALSE) {
        stop("Invalid output_type: choices are performance or rfobject")
    }
    # Set up our data into a fully expanded edgelist
    require(tidyr)
    dat$real <- 1  #make a new column denoting all of these edges are real; important when we expand ou
    dat <- dplyr::filter_at(dat, vars(c(FrugTraits, PlantTraits)),
        all_vars(!is.na(.)))  #make sure we have data for all our predictors

    dat <- dplyr::select_at(dat, vars(c(Frugivore_Species, Plant_Species,
        real, FrugTraits, PlantTraits)))  #Select our relevant predictors
    dat %<>%
        unique(.)  #Make sure we only have unique entries
    full_L <- tidyr::expand(dat, Frugivore_Species, Plant_Species)  #Expand to include all possible pai
    full_real <- dplyr::select(dat, Frugivore_Species, Plant_Species,
        real) %>%
        left_join(full_L, ., by = c("Frugivore_Species", "Plant_Species"))  #notating which of our edge
    full_real$real[is.na(full_real$real) == TRUE] <- 0

    full_real_frugs <- dat[, c("Frugivore_Species", FrugTraits)] %>%
        unique() %>%
        left_join(full_real, ., by = "Frugivore_Species")  #Add in our frugivore traits

    full_real_both <- dat[, c("Plant_Species", PlantTraits)] %>%
        unique() %>%
        left_join(full_real_frugs, ., by = "Plant_Species")  #add in our plant traits

    full_real_both$real <- as.factor(full_real_both$real)

    rf <- randomForest::randomForest(real ~ . - Frugivore_Species -
        Plant_Species, data = full_real_both, ntree = 100)
    # ROC <- roc(full_real_both$real, rf$votes[,2])
    predictions <- stats::predict(rf, newdata = full_real_both,
        type = "prob")

    if (output_type == "predictions") {
        output <- full_real_both %>%
            select(., "Frugivore_Species", "Plant_Species", real)
        output$S <- predictions[, 2]
        return(output)
    }

    if (output_type == "rfobject") {
        output <- rf
        return(output)
    }
}
```

Next, we create a situation where empirical networks using the generation process outlined above. We also here define a sampling rate, specifying the number of links observed in addition to the number of true links.

We use singular value decomposition on the osberved network, and use those features to inform a random forest model trained on the observed positive interactions and enough background points to create a 25%

prevalence. We evaluate this model through AUC on the held-out positive links as well as enough new background points to again acheive a 25% prevalence on the test set.

```
# n_plants <- 100; n_frugivores=50; n_links = 500;
# n_observed = 300; backgroundratio=3 #for debugging
run_latent_trait_skewsamp_simulation <- function(n_plants = 100,
    n_frugivores = 50, n_links = 500, n_observed = 300, backgroundratio = 3,
    rndm = FALSE, aundweightsamp = T) {
    # Generate all combinations
    interactions <- expand.grid(FrugivoreID = 1:n_frugivores,
        PlantID = 1:n_plants) %>%
        left_join(data.frame(FrugivoreID = 1:n_frugivores, FrugAbund = rlnorm(n_frugivores)),
            by = "FrugivoreID") %>%
        left_join(data.frame(PlantID = 1:n_plants, PlantAbund = rlnorm(n_plants)),
            by = "PlantID") %>%
        mutate(AbundProd = PlantAbund * FrugAbund)  #Make a column of abundance products
    # Assign 'real' interactions using abundance-weighted
    # probabilities
    interactions$real <- 0
    interactions$real[sample(1:nrow(interactions), size = n_links,
        replace = FALSE, prob = interactions$AbundProd)] <- 1

    interactions$obs <- NA

    Y <- xtabs(real ~ FrugivoreID + PlantID, data = interactions)  #Make adj matrix
    Y

    if (aundweightsamp == T) {
        # Degrade the network by hiding some links AGAIN
        # ABUNDANCE SAMPLING
        interactions$obs[sample(which(interactions$real == 1),
            size = n_observed, replace = FALSE, prob = interactions$AbundProd[which(interactions$real ==
                1)])] <- 1  #Correctly get n_observed BASED ON ABUND
    }
    if (aundweightsamp == F) {
        # Degrade the network by hiding some links RANDOM
        # SAMPLING
        interactions$obs[sample(which(interactions$real == 1),
            size = n_observed, replace = FALSE)] <- 1  #Correctly get n_observed BASED ON ABUND
    }

    interactions$obs[is.na(interactions$obs)] <- 0
    # Rename species
    interactions$FrugivoreID <- paste0("F", interactions$FrugivoreID)
    interactions$PlantID <- paste0("P", interactions$PlantID)
    # Perform SVD latent feature reduction
    mat_assym <- xtabs(obs ~ FrugivoreID + PlantID, data = interactions)  #Make adj matrix
    decomp <- svd(mat_assym)
    u <- decomp$u
    v <- data.frame(decomp$v)
    # Trait tables
    plantsSVD <- data.frame(PlantID = colnames(mat_assym), Psvd1 = v[,
        1], Psvd2 = v[, 2], Psvd3 = v[, 3])
    frugSVD <- data.frame(FrugivoreID = rownames(mat_assym),
```

```
        Fsvd1 = u[, 1], Fsvd2 = u[, 2], Fsvd3 = u[, 3])
    # Merge traits
    interactions <- interactions %>%
        left_join(plantsSVD, by = "PlantID") %>%
        left_join(frugSVD, by = "FrugivoreID")
    # Create test/training split
    test <- filter(interactions, real == 1 & obs == 0)  #Test is all unobserved real
    background <- filter(interactions, real == 0)
    test <- bind_rows(test, background[sample(nrow(background),
        backgroundratio * nrow(test)), ])

    # Add in some other missed 0's as a background ratio
    test$combo <- paste(test$FrugivoreID, test$PlantID, sep = "-")  #create unifying column
    interactions$combo <- paste(interactions$FrugivoreID, interactions$PlantID,
        sep = "-")
    train <- filter(interactions, !combo %in% test$combo)  #training is everything left (should this be
    # Rename for woodedWalk
    train <- rename(train, Frugivore_Species = FrugivoreID, Plant_Species = PlantID)
    test <- rename(test, Frugivore_Species = FrugivoreID, Plant_Species = PlantID)

    if (rndm == T) {
        test$real <- sample(test$real, length(test$real), replace = F)
    }
    # Fit model based on latent traits
    out <- woodedWalk_NoSplit(train, FrugTraits = c("Fsvd1",
        "Fsvd2", "Fsvd3"), PlantTraits = c("Psvd1", "Psvd2",
        "Psvd3"), class_balancing = TRUE)

    test$S <- stats::predict(out, newdata = test, type = "prob")[,
        2]  #Grab Svals

    # Evaluate performance
    rOC <- roc(data = test, response = real, predictor = S)
    AUC <- rOC$auc
    return(AUC)
}
```

Below, we simulate this process across a variety of a) graph sizes, b) connectance values of the "true" graph, c) sampling completeness values, in order to characterize the potential of latent traits for prediction across this parameter space. This neutral case represents the absolute simplest case in which latent features may be useful. The only factor impacting species' probabilities to interact are their abundances, while in real systems there may be many unmeasured factors impacting interaction structures which latent features may be able to detect. It is however a useful investigation into how latent features might change across this parameter space.

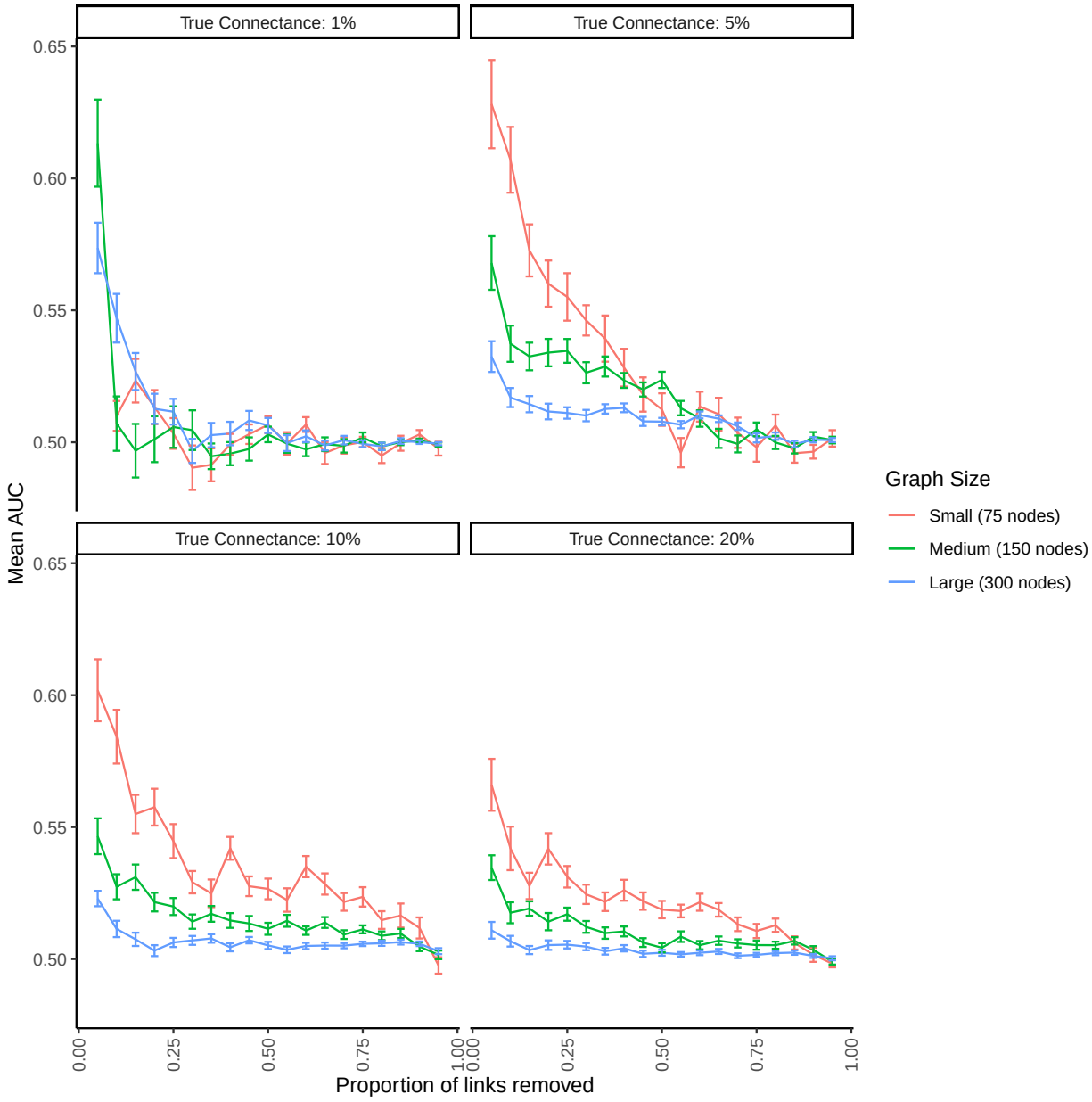

Figure S9: Plot of mean decrease in AUC performance as a function of the proportion of links removed from the observed set. Plot facets represent alternative connectance values of the true graph; color represent alternative graph sizes. Aside from the extreme sparse case (connectance = 1%), latent features tended to perform best on the smallest graphs. Performance overall was highest on moderately sparse graphs (5% connectance); across all connectance values and graph sizes, model performance decreases with greater sample degredation.

References

Gotelli N.J. (2000) Null model analysis of species co-occurrence patterns. Ecology, 81, 2606-2621

Vázquez, D. P., Blüthgen, N., Cagnolo, L., & Chacoff, N. P. (2009). Uniting pattern and process in plant–animal mutualistic networks: a review. Annals of botany, 103(9), 1445-1457.