# SUPPLEMENTARY DATA

## Variance of a single community-weighted mean decomposition rate ($K_M$)

We begin with the same distributional assumption as the mixed-model regression (Model 1), namely that the decomposition rates of the species in this study are drawn from a normally distributed statistical population consisting of a large set of species in which each species has a particular monoculture decomposition rate, $k_i$; i.e. $k \sim N\ (\mu,\sigma_\delta)$. The population mean ($\mu$) and standard deviation ($\sigma_\delta$) of decomposition rates, estimated in the mixed-model monoculture regression, were –2.49×10$^{-3}$ and 6.80×10$^{-4}$, respectively. We randomly choose $S$ (here, 6) species from this statistical population and then construct a subset M of these $S$ species, consisting of a total of $S_M$ species from this larger set; the subset M represents the subset of species forming a particular litter mixture. We calculate a weighted mean (a "community-weighted mean") of the resulting $S_M$ values of $k_i$. In other words, let $K_M = \sum_{i \in M}^{S_M} p_i k_i$ be the community weighted mean of the mixture decomposition rate in the set of species belonging to a particular mixture M. Here, p = {$p_1$,$p_2$,…,$p_{S_M}$} are the proportions of each species in the mixture.

A community-weighted mean (CWM) as defined here is a weighted sum of the random variables $k_i$. The general formula for the variance of a weighted sum of random variables is given in equation 1 (for example, Freund, 1962).

$$Var(K_M) = \sum_{i \in M}^{S_M} p_i^2 Var(k_i) + 2\sum_{i \in M}^{S_M} \sum_{j \in M \& j > i}^{S_M} p_i p_j Cov(k_i, k_j) \qquad \text{Eqn 1}$$

However, in our case all of the $k_i$ come from the same distribution and so Var($k_i$)= $\sigma_\delta^2$ and the first term on the RHS of equation 1 in our experiment is $\left(\frac{1}{S_M^2}\right)\sum_{i=1}^{S_M} Var(k_i) = \left(\frac{1}{S_M^2}\right) S_M \sigma_\delta^2 = \frac{\sigma_\delta^2}{S_M}$ . Since the litter mixtures were formed for all possible unique combinations of each species irrespective of the decomposition rates of each ($k_i$), the probability of species $j$ being a member of the mixture is the same whether or not species $i$ has already been chosen or not, and so Cov($k_i$, $k_j$)=0. This would not be true in a sample of naturally occurring litter because the presence/absence of species can be correlated. Therefore, in our experiment:

$$Var(K_M)=\sum_{i=1}^{i\in M} p_i^2 Var(k_i)=\frac{\sigma_\delta^2}{S_M}=\frac{0.00057^2}{S_M}$$

**Covariances**

The different mixtures having the same total number of species formed different, but overlapping pools of these subsets. For instance, given a set of 3 species (species 1, 2, 3), one can form a total of three different subsets (i.e. mixed-species litters) of two species {(1,2), (1,3), (2,3)} and calculate three different CWM ($K_{12}$, $K_{13}$, $K_{23}$). Each CWM is a new random variable. However, each pair of subsets in this example shares one species in common and this will generate a positive covariance between each pair of CWM. The general formula for the covariance between two random variables that are each weighted sums (i.e. CWM) is given in equation 2 (e.g. Freund, 1962).

$$Cov(K_{M_1},K_{M_2})=\sum_{i\in M_1}\sum_{j\in M_2} p_i p_j Cov(k_i,k_j)=\sum_{i\in\{M_1\cup M_2\}} p_k^2 Var(k_i)+\sum_{i\notin\{M_1\cup M_2\}}\sum_{j\notin\{M_1\cup M_2\}} p_i p_j Cov(k_i,k_j) \quad \text{Eqn 2.}$$

The first term on the RHS of equation 2 follows from the fact that the covariance of a random variable with itself is the variance of this random variable, summed over all species that are present in both mixtures $M_1$ and $M_2$. The second term on the RHS sums over all species that are not present in both mixtures $M_1$ and $M_2$. For instance, using the above CWM values,

$Cov(K_{12},K_{13})=p_1^2 Var(k_1)+p_1p_3Cov(k_1,k_3)+p_2p_1Cov(k_2,k_1)+p_2p_3Cov(k_2,k_3)$ since species 1 is present in both mixtures, while species 2 and 3 are not present in both mixtures. Again, the covariances between the monoculture *k* values will be zero in our experiment, and so $Cov(K_{12},K_{13})=p_1^2 Var(k_1)$. In general, if there are no species in common between the two mixtures, then the covariance between their CWMs will be zero. Every time the same species appears in two different mixtures, this generates a non-zero covariance. Given any pair ($M_1$, $M_2$) of mixtures, each containing the same total number ($S_M$) of species,

$Cov(K_{M_1},K_{M_2}) = \sum_{a=1}^{C} p_a^2 Var(k_a)$ where C is the number of species in common between mixtures $M_1$ and $M_2$.

Since the litter mixtures always had the same initial amount of litter per species, in our experiment

$Cov(K_{M_1},K_{M_2}) = \dfrac{CVar(k)}{S_M^2}$.

Given a total of *N* different mixtures, each possessing a total of $S_M$ species, we can now specify the variance-covariance matrix, Σ, of our *N* unique *K* values, where $C_{ij}$ is the number of species in common between mixtures *i* and *j*, and $\sigma_\delta^2$ [here, $(6.80\times10^{-4})^2$] is the interspecific variance of the monoculture decomposition rates:

$$\sum = \begin{bmatrix} \dfrac{\sigma_\delta^2}{S_M} & \dfrac{C_{12}\sigma_\delta^2}{S_M^2} & \cdots & \dfrac{C_{1N}\sigma_\delta^2}{S_M^2} \\ \dfrac{C_{21}\sigma_\delta^2}{S_M^2} & \dfrac{\sigma_\delta^2}{S_M} & \cdots & \dfrac{C_{2S}\sigma_\delta^2}{S_M^2} \\ \cdots & \cdots & \cdots & \cdots \\ \dfrac{C_{N1}\sigma_\delta^2}{S_M^2} & \dfrac{C_{N2}\sigma_\delta^2}{S_M^2} & \cdots & \dfrac{\sigma_\delta^2}{S_M} \end{bmatrix} \qquad \text{Eqn 3}$$

Clearly, Σ tends towards a diagonal matrix with diagonal values of $\sigma_\delta{}^2/S_M$ as the number of species in common in mixtures *i* and *j* decreases. As the number of species in common ($C_{ij}$) increases, the strength of the positive covariances between the *K* values also increases and the differences between these *K* values will decrease.

## The distribution of $K_M$ values in our experiment

Given the assumption that, in common with the mixed-model regression, the monoculture decomposition values (*k*) are approximately normally distributed, the CWM ($K_M$) values will follow an *N*-dimensional multivariate normal distribution the covariance matrix of which is Σ. The expected value of $K_M$ is $E\left[K_M\right] = E\left[\sum_{i=1}^{i\in P_M} p_i k_i\right] = \sum_{i=1} p_i E\left[k_i\right]$. In our experiment, this reduces to $\dfrac{1}{S}\sum_{i=1}^{i\in P_M} E\left[k_i\right]$. However, the $k_i$ values are random draws from the same interspecific population of decomposition rates and so the expected values of $k_i$ are simply the mean of this interspecific population, i.e. $\mu$ (here, $-2.49\times10^{-3}$). Therefore, the distribution of the $K_\mathrm{M}$ values is multivariate normal with mean vector $\{\mu,\mu,\ldots,\mu\}$ and covariance matrix $\Sigma$. The `mvrnorm` function in the *MASS* library of R generates random multivariate vectors given the mean vector and the covariance matrix. A single draw from this multivariate normal distribution represents a vector of CWMs for each unique set of mixtures having the chosen number of species and the standard deviation of these values quantifies the degree of variation between them. By generating a large number of independent draws from this multivariate normal distribution and calculating the standard deviation of the values in each independent vector, one can approximate the sampling distribution of these standard deviations. The vertical lines in Figure 1A show the 95% confidence intervals of this sampling distribution. By comparing the standard deviation of the actual CWM with this sampling distribution, one can estimate the probability that the observed standard deviation comes from this sampling distribution. This is repeated for mixtures having 2, 3 or 5 species per mixture. This Monte Carlo procedure is implemented in the R function “simulate.mixtures” :

```
function (mean.k, var.k, total.number.species, species.per.mixture,
    n.sim = 20000, simulate.infinite.species.pool = T, measured.sd.k)
{
# This function takes the mean and variance of the monoculture decomposition rates, as estimated
by a mixed model regression, and calculates the covariance matrix of CWM decomposition rates of
each possible mixture of size "species.per.mixture" given a total of "total.number.species" in
the pool, assuming biomass-ratio hypothesis (BMRH). "n.sim" : number of independant simulations
used to estimate the distribution of these mixture decomposition rates.
# The test statistic is the SD of the k-values between mixtures having the save
species.per.mixture
# "measured.sd.k" : estimate the probability that the SD of your measured values. H0 : the
measured values comes from the simulated distribution. H1 : the SD of the measured values is
smaller than expected from the simulated distribution.
if (total.number.species <= species.per.mixture)
        stop("Error. species per mixture\n must be less than the total number of species")
    combos <- combn(total.number.species, species.per.mixture)
    ncombos <- dim(combos)[2]
# Cs : covariance matrix of mixtures
# diagonals : var.k/species.per.mixtures
# off-diagonals : species.in.common*var.k/(species.per.mixture)^2
    Cs <- matrix(0, ncombos, ncombos)
    for (i in 1:ncombos)
        {
        for (j in i:ncombos)
            {
            # when i=j (i.e. a diagonal), unique.species=# species in the mixture
            unique.species <- length(unique(c(combos[, i], combos[, j])))
            species.in.common <- 2 * species.per.mixture - unique.species
            Cs[i, j] <- Cs[j, i] <- var.k * species.in.common/(species.per.mixture^2)
            }
        }
    if (simulate.infinite.species.pool)
        {
        Cs2 <- matrix(0, ncombos, ncombos)
        diag(Cs2) <- var.k/species.per.mixture
        }
    library(MASS)
    x <- mvrnorm(n = n.sim, mu = rep(mean.k, ncombos), Sigma = Cs)
    if (!simulate.infinite.species.pool)
       {
        min.K <- apply(x, 1, min)
        max.K <- apply(x, 1, max)
        mean.K <- apply(x, 1, mean)
        sd.K <- apply(x, 1, sd)
        null.prob <- sum(measured.sd.k >= sd.K)/n.sim
        out <- list(median.min.K = median(min.K), median.max.K = median(max.K),
            median.sd.K = median(sd.K), quantiles.sd.K = quantile(sd.K,
            probs = c(0.025, 0.5, 0.975)), prob.SD.LE.simulation = null.prob)
        hist(sd.K, xlab = "SD(K)", main = paste("N=", as.character(n.sim),
            "simulations", "p=", as.character(round(null.prob, 3))))
        limits <- quantile(sd.K, probs = c(0.025, 0.975))
        lines(x = rep(limits[1], 2), y = c(0, 0.25 * length(sd.K)))
        lines(x = rep(limits[2], 2), y = c(0, 0.25 * length(sd.K)))
        if (!is.na(measured.sd.k))
            lines(x = rep(measured.sd.k, 2), y = c(0, 0.25 *
                length(sd.K)), lwd = 2)
        }
    if (simulate.infinite.species.pool) {
        x.inf <- mvrnorm(n = n.sim, mu = rep(mean.k, ncombos),
            Sigma = Cs2)
        min.K <- apply(x, 1, min)
        max.K <- apply(x, 1, max)
        mean.K <- apply(x, 1, mean)
        sd.K <- apply(x, 1, sd)
        null.prob <- sum(measured.sd.k >= sd.K)/n.sim
        sd.K.inf <- apply(x.inf, 1, sd)
        out <- list(median.min.K = median(min.K), median.max.K = median(max.K),
            median.sd.K = median(sd.K), quantiles.sd.K = quantile(sd.K,
            probs = c(0.025, 0.5, 0.975)), median.sd.K.infinite.pool =
            median(sd.K.inf), quantiles.sd.K.inf = quantile(sd.K.inf,
            probs = c(0.025, 0.5, 0.975)), prob.SD.LE.simulation = null.prob)
        par(mfrow = c(2, 1))
        hist(sd.K, xlab = "SD(K)", main = paste("N=", as.character(n.sim),
            "simulations", "p=", as.character(round(null.prob, 3))))
        limits <- quantile(sd.K, probs = c(0.025, 0.975))
        lines(x = rep(limits[1], 2), y = c(0, 0.25 * length(sd.K)))
        lines(x = rep(limits[2], 2), y = c(0, 0.25 * length(sd.K)))
        if (!is.na(measured.sd.k))
            lines(x = rep(measured.sd.k, 2), y = c(0, 0.25 * length(sd.K)), lwd = 2)
            hist(sd.K.inf, xlab = "SD(K)", main = "assuming infinite species pool")
            limits <- quantile(sd.K.inf, probs = c(0.025, 0.975))
            lines(x = rep(limits[1], 2), y = c(0, 0.25 * length(sd.K.inf)))
            lines(x = rep(limits[2], 2), y = c(0, 0.25 * length(sd.K.inf)))
        if (!is.na(measured.sd.k))
            lines(x = rep(measured.sd.k, 2), y = c(0, 0.25 * length(sd.K)), lwd = 2)
        par(mfrow = c(1, 1))
    }
    out
}
```

## Testing the significance of the observed decrease in the prediction errors with increasing species richness of the litter mixture

Figure 1B shows the observations of the deviation ($d_{ij}$) between the observed and predicted (community-weighted) decomposition rates of mixture *i* in the set of mixtures having a total of $j$=2,3,5 species. The alternative hypothesis is that the variability of these prediction errors decreases as the number of species in the mixture increases. The null hypothesis is the contrary. Given this null hypothesis that the variability of the prediction errors is independent of the species richness of the mixture, we can test the null hypothesis using a permutation test (Manly, 1997) as follows.

1. We calculate the standard deviation ($S_j$) of the observed $d_{ij}$ values between the different combinations of mixtures for each fixed number ($j$) of species.
2. We regress $S_j$ on $j$ in a simple linear regression and obtain the slope that describes by how much the variation in the prediction errors decreases with each unit increase in species richness.
3. We randomly permute the $d_{ij}$ values between the three levels of species richness while maintaining the same number of values per level as in the original data set.
4. We repeat steps 1–3 many ($N$) times.
5. We count the number ($n$) of times that the slope of these permuted values is more negative than the observed value.

The 1-tailed null probability of observing at least the observed amount of decrease in the variability of the prediction errors by chance, given the null hypothesis, is estimated by 1–$p$, where $p = n/N$. If $p < 0.05$ then the null hypothesis is rejected and the alternative is accepted (*i.e.*, the values of $d_{ij}$ decrease with increasing j). The R function "permute.errors" implements this permutation test :

```
function (percent.error = T, n.perms = 1000)
{
    datas<-read.table("/…/datas.txt", header = T)
    datas_t_temp<-Tk_dec_2009_4[datas$type==1,]
    datas_t<-Tk_dec_2009_4_t_temp[datas_t_temp$nb_sp!=1,]
    if (!percent.error)
        error.k <- 1000 * (datas_t$k_obs - datas_t$k_CWM)
    if (percent.error)
        error.k <- 100 * (datas_t$k_obs - datas_t$k_CWM)/ datas_t$k_obs
    Nspecies <- datas_t$nb_sp
    Nobs.per.Nspecies <- rep(NA, 3)
    x <- c(2, 3, 5)
    for (i in 1:3)
    {
        Nobs.per.Nspecies[i] <- length(Tk_dec_2009_4_t$nb_sp[Tk_dec_2009_4_t$nb_sp == x[i]])
    }
    sd.data <- matrix(NA, nrow = 20, ncol = 3)
    for (i in 1:3) sd.data[1:Nobs.per.Nspecies[i], i] <- error.k[Nspecies == x[i]]
    slope.sd <- coef(lm(apply(sd.data, 2, sd, na.rm = T) ~ x[1:3]))[2]
    perm.slopes <- rep(NA, n.perms + 1)
    for (p in 1:n.perms) #boucle pour les permutations
    {
        for (i in 1:3) sd.data[1:Nobs.per.Nspecies[i], i] <- sample(error.k)[Nspecies == x[i]]
        perm.slopes[p] <- coef(lm(apply(sd.data, 2, sd, na.rm = T) ~ x[1:3]))[2]
    }
    perm.slopes[n.perms + 1] <- slope.sd
    par(mfrow = c(1, 1))
    hist(perm.slopes, xlab = "Slope SD~Nspecies")
    prob <- sum(perm.slopes <= slope.sd)/n.perms
    ci95<-1.95*sqrt(prob*(1-prob)/n.perms)
    lines(x = c(slope.sd, slope.sd), y = c(0, 0.25 * n.perms), lwd = 2)
    list(observed.slope = slope.sd, null.prob = prob, ci95=c(prob-ci95,prob+ci95))
}
```

## REFERENCES

**Freund JE. 1962.** *Mathematical statistics*, Englewood Cliffs, NJ, Prentice-Hall.

**Manly BFJ. 1997.** *Randomization, Bootstrap and Monte Carlo Methods in Biology, second edition*, London, Chapman and Hall.