Supplementary Data

Table S1) Vendor and price for each component used for the proposed drone sampling method.

Equipment	Vendor	Price (JPY)	GBP (£)
DJI Phantom 3 Standard x 1	Mercari	40,000	206.16
Poweregg X x 1	Mercari	25,000	131.10
7.4v Battery x 2	Amazon	2,599	13.40
FlextailGear Tiny Pump x 1	Amazon	2890	14.90
25mm Filter holder x 10	Amazon	14157	72.97
25mm Filter x 100	Monotaro	4290	22.11
Arduino Nano x 3	Amazon	4,914	25.33
Arduino UNO R3	Mercari	1,800	9.26
Servo SG90 x 5	Amazon	999	5.15
3M Command Strips	Amazon	894	4.61
Plastic Black Junction box x 1	Amazon	335	1.73
Jumper Wire Set x 1	Amazon	759	3.91
Universal PCB 90mm x 150mm	Mercari	850	4.37
Universal PCB 30mm x 30mm	Amazon	650	3.34
Total		**100,137**	**518.07**

Table S2) Weight of the combined components for the DJI Phantom 3 Standard set up.

Component	Weight (Grams)
Pump + 25mm Filter Holder + SG90 Servo	101.45
Lipo battery 7.4v 1300Amh +Arduino UNO R3 + Breadboard (90mm x 150mm)	107.7
Total Weight	**209.15**

Table S3) Weight of the combined components for the Poweregg X set up.

Component	Weight (Grams)
Pump + 25mm Filter Holder + SG90 Servo	101.45
Lipo battery 7.4v 800Amh	46
Arduino Nano	7
Breadboard (30mm x 30mm)	2.35
Plastic Black Junction box (100mm x 60mm x 25mm)	36.38
Total Weight	**193.18**

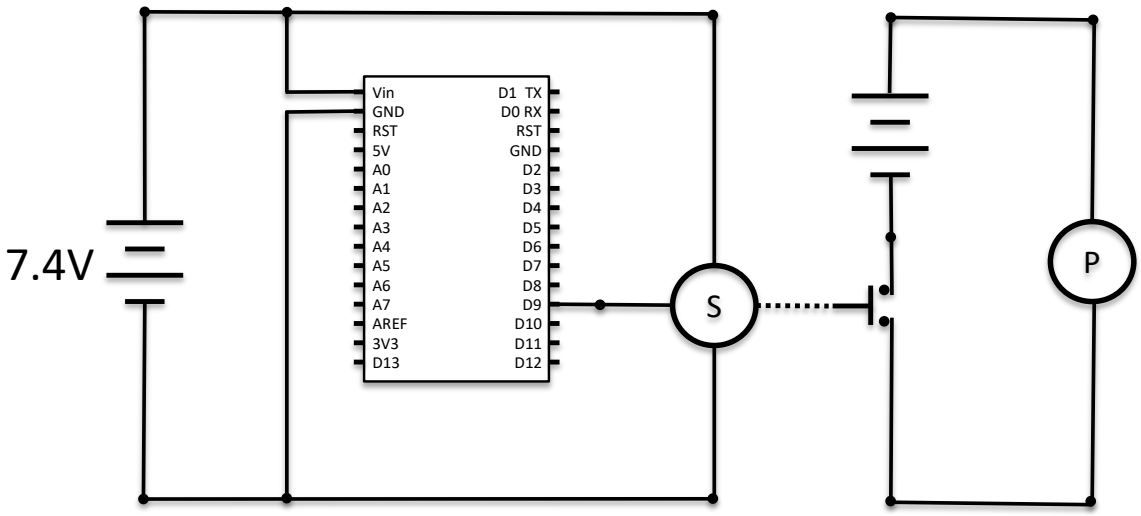


Fig. S1) Basic schematic of the wiring between the 7.4v battery, Arduino and Servo (S). The circuit is wired in parallel.

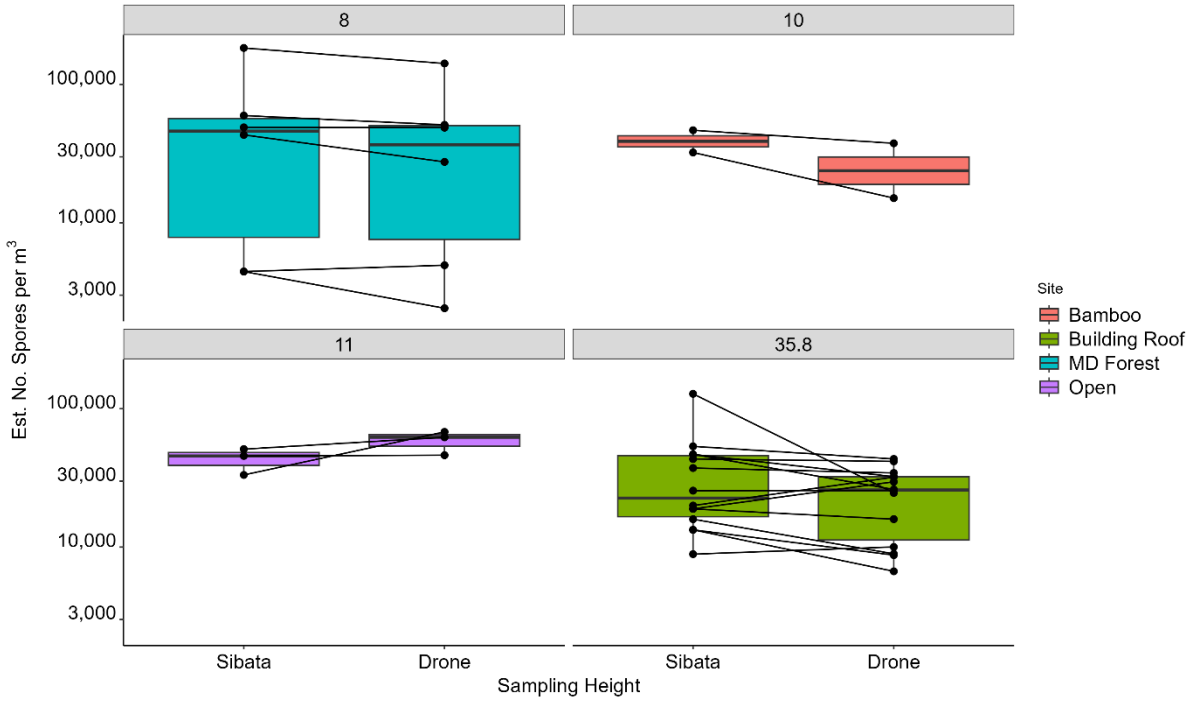


Fig. S2) Breakdown of comparisons in each habitat sampled at the same height between Sibata pump and Drone pump, numbers in grey bars indicate the sampling height in metres *(n = 25, dry = 20, rain = 5)*.

Fig. S3) Breakdown of comparisons between Sibata pump (Ground) and Drone sampling above the canopy in each habitat *(n = 16, dry = 15, rain = 1)*.

Equation. S1) Equations used for Spore quantification.

The microscope used for quantification was a Nikon Optiphot (Nikon, Japan).

In order to estimate total spore counts per filter the following equation was used;

$$A_{\mathrm{FOV}} = \pi\, r_{\mathrm{FOV}}^2$$

$$A_{\mathrm{f}} = \pi\, r_{\mathrm{f}}^2$$

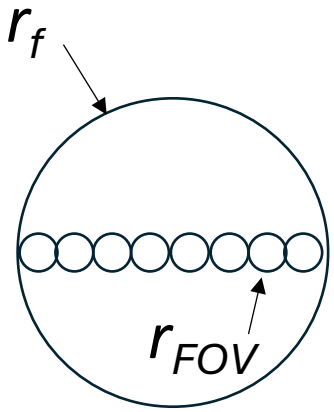
$$A_{\mathrm{tran}} = n_{\mathrm{FOV}} \times A_{\mathrm{FOV}}$$

$$n_{\mathrm{t,\ FOV}} = {}^{A_{\mathrm{f}}}\!/_{A_{\mathrm{FOV}}}$$

$$\bar{n}_{\mathrm{spore,FOV}} = {}^{n_{\mathrm{t\ spore,FOV}}}\!/_{n_{\mathrm{FOV}}}$$

$$n_{\mathrm{t,FOV}} = {}^{A_{\mathrm{f}}}\!/_{A_{\mathrm{FOV}}}$$

$$n_{\mathrm{t\ spores}} = \bar{n}_{\mathrm{spore,FOV}} \times n_{\mathrm{t,FOV}}$$



Abbreviations

A_{FOV} - Area of 1 FOV (Field of view)

r_{FOV} - Radius of 1 FOV

A_{f} - Area of GF filter

r_{f} - Radius of GF filter

A_{tran}- Area of transect (FOV's counted across the diameter of filter)

n_{FOV} - Number of counted FOV along the transect

$n_{\mathrm{t,FOV}}$ - Total number of FOV's in 1 GF filter

$n_{\mathrm{t\,spore,FOV}}$ - Total number of counted spores in n_{FOV}

$\bar{n}_{\mathrm{spore,FOV}}$ - Mean number of spores of transect FOV's.

$n_{\mathrm{t\,spores}}$ - Estimated number of spores in filter

Equation. S2) Estimated number of spores per m^3 of air.

$$V_{\mathrm{air}} = t \times Q$$

$$D_{\mathrm{spores}} = \left({}^{n_{\mathrm{t\,spores}}}/_{V_{\mathrm{air}}}\right) \times 10^3$$

t - sampling duration (min)

V_{air} - total volume of air (L)

Q - flow rate (L/min)

D_{spores} - density of spores (num/m^3)

Programming code S1) Code used for the Arduino (Nano and UNO R3) to program the ServoSG90 to press the on and off button for the pump allowing for a timed system. The yellow highlighted parts are waiting times. 60000ms is a 1-minute delay before the servo presses the on button of the pump. 600000ms is 10 minutes of sampling time, after 10 minutes the servo will switch off the pump.

```
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  Serial.begin(9600);
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  for (pos = 0; pos <= 0; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(60000);                       // waits 60s for the servo to reach the position
  }for (pos = 0; pos <= 35; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
    for (pos = 35; pos <= 35; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(100);                       // waits 100ms for the servo to reach the position
  }
  for (pos = 35; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
   myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  for (pos = 0; pos <= 0; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(100);                       // waits 100ms for the servo to reach the position
  }for (pos = 0; pos <= 35; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
    for (pos = 35; pos <= 35; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
  for (pos = 35; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
    myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  for (pos = 0; pos <= 0; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(600000);                       // waits 10minutes for the servo to reach the
position
  }for (pos = 0; pos <= 35; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
    for (pos = 35; pos <= 35; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(300);                       // waits 300ms for the servo to reach the position
  }
  for (pos = 35; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(10);                       // waits 10ms for the servo to reach the position
  }
}
void loop() {}
```