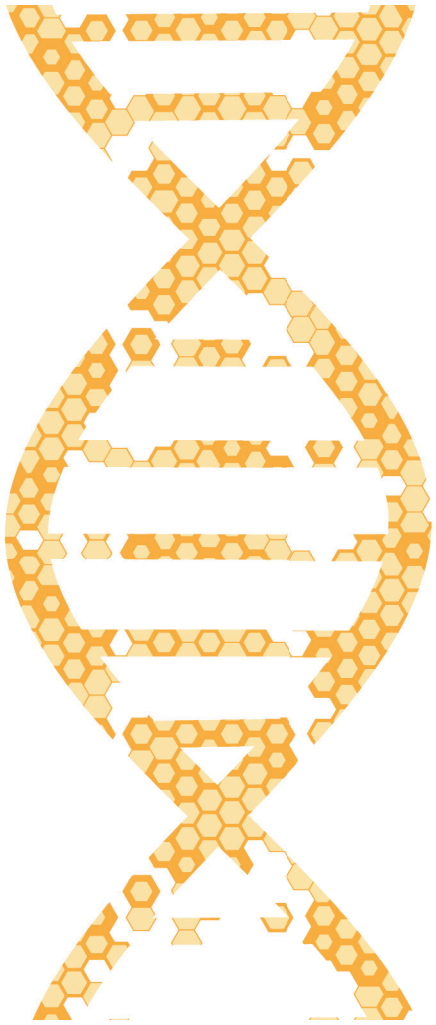
BTI Institute

Borders • Trade • Immigration

A Department of Homeland Security Center of Excellence

DNA Assays for Determining Honey Origins



Project Report

Released 25 June 2021

The Borders, Trade, and Immigration Institute

A Department of Homeland Security Center of Excellence
Led by the University of Houston

Thank You

This product, along with everything we do, is dedicated to the men and women of the United States Department of Homeland Security. We thank them for their tireless efforts to secure our Nation and safeguard our economic prosperity by facilitating lawful travel and trade.

Contact
Email: bti@uh.edu
Website: www.uh.edu/bti/
Twitter: @bti_uh
LinkedIn: Borders, Trade, and Immigration

Final Report

DNA Assays for Determining Honey Origins

Dr. Richard C. Willson, University of Houston (PI)
Dr. Aniko Sabo, Baylor College of Medicine
Dr. Katerina Kourentzi, University of Houston

Graduate students:
Dimple Chavan, University of Houston (Graduate student)
Dr. Jay R T. Adolacion, University of Houston (Former Willson Lab student)
Suman Nandy, University of Houston (Graduate student)

Other personnel:
Najia Sherwani (Undergraduate student)
Mehrnoosh Kohansal (Lab Technician)

EXECUTIVE SUMMARY

Adulteration and mislabeling of honey to mask its true origin have become a global issue. Pollen microscopy, the current gold standard for identifying the geographical origins of honey, is time-consuming and requires expert personnel. Additionally, pollen microscopy cannot source honey samples that have been filtered to remove the original pollen and/or spiked with pollen from more-remunerative plants.

In this work we explored the DNA-based characterization of honey origins using deep sequencing targeting the nuclear ribosomal Internal Transcribed Spacer 2 (ITS2) spacer DNA between the small- and large-subunit ribosomal RNA genes of plant genomic DNA, known to facilitate species-level discrimination of plants. Using next-generation sequencing (NGS) and clustering analysis, we have assembled country-specific plant DNA sequences obtained from NGS of plant genomic DNA isolated from 300 honey samples. We also have successfully isolated trace DNA and sequenced plant ITS2 from pollen-free, filtered honey using three methods: (i) anti-dsDNA antibodies coupled to magnetic particles; (ii) batch adsorption on Q Sepharose anion exchanger; and (iii) batch adsorption on ceramic hydroxyapatite. The amplified ITS2 region of the captured pollen-free DNA was sequenced using next-generation sequencing and was found to be identical to plant ITS2 of pollen DNA from the same honey sample. Enrichment of trace pollen-free DNA from filtered honey samples opens a new approach to identify the true origins of filtered honey samples, and may suggest other applications of DNA-based product sourcing.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STBTI00001-03-00 (formerly 2015-ST-061-BSH001). The authors gratefully acknowledge the participation of the DHS Project Champion, Deputy Executive Director Patricia Hawes-Coleman, BTI staff, and the members of the BTI Research Committee, including George Zouridakis Ph.D., Luca Pollonini Ph.D., and Elaine Liu Ph.D.

The authors would also like to thank members of the Willson lab and other University of Houston colleagues for their aid in acquiring three hundred honey samples from various countries before and during the COVID-19 pandemic.

DISCLAIMER: The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security or the University of Houston.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

DNA	Deoxyribonucleic acid
gDNA	Genomic DNA
ITS2	Internal transcribed spacer 2
PCR	Polymerase Chain Reaction
NGS	Next-generation sequencing
dsDNA	Double stranded DNA
Ab	Antibody
CHT Type I	Ceramic Hydroxyapatite type I
PES	Polyethersulfone

1. INTRODUCTION

U.S. honey production has remained stable over the past ten years averaging to about 157 million pounds as reported by United States Department of Agriculture.[1] However, U.S. honey imports have increased tremendously from 251 million pounds in 2010 to nearly 416 million pounds in 2019 (Fig 1). Since 2006, imported honey accounts for majority of U.S. honey supplies. Over the recent years, imported honey has primarily been obtained from Argentina, Vietnam, and India, which accounted for 65 percent of total honey imports in 2019 (Fig 2).

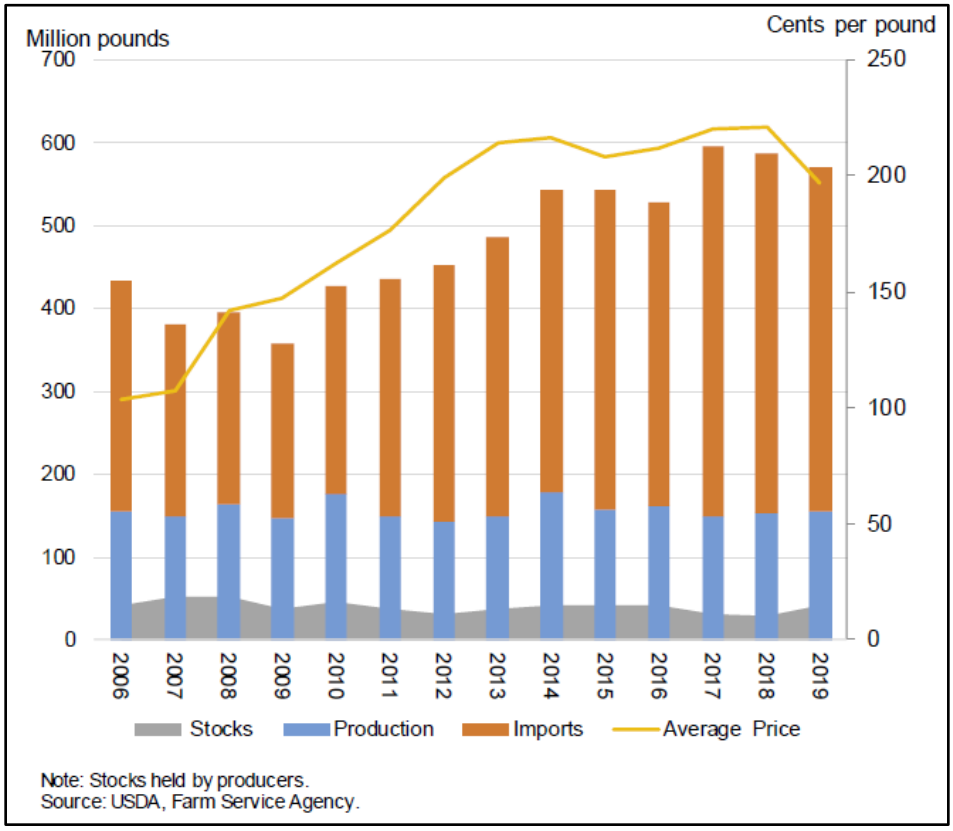


Figure 1. U.S. honey supplies and prices from 2006 to 2019.

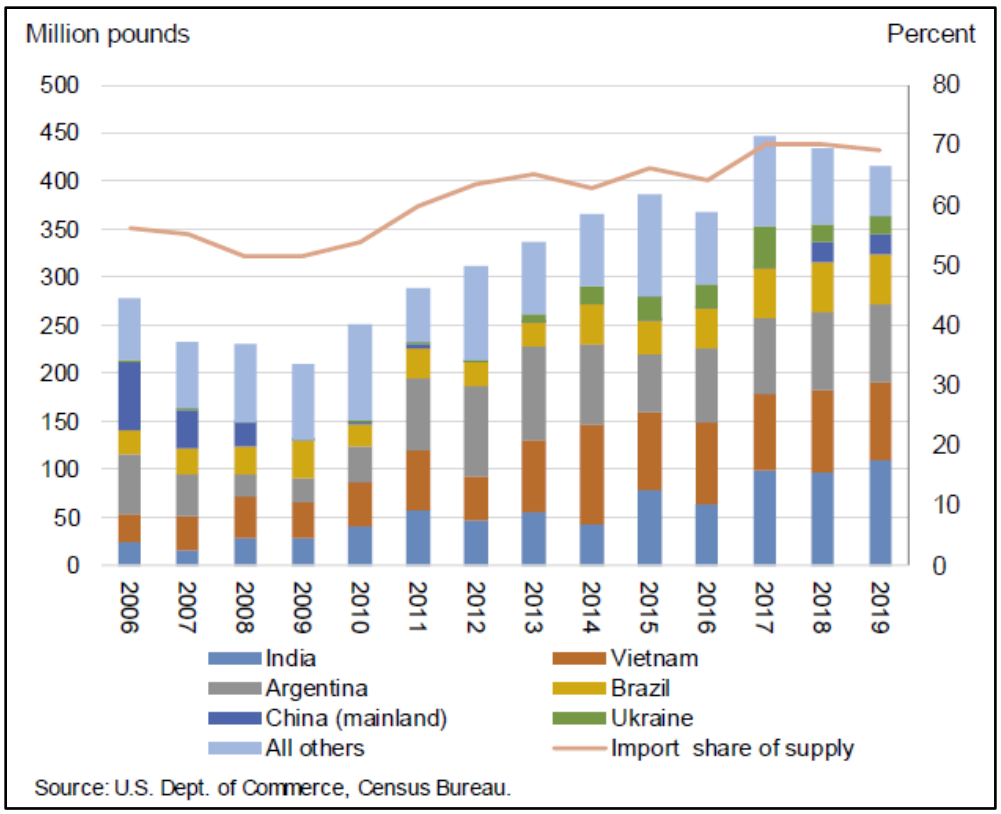


Figure 2. U.S. honey imports by country and import share of total supplies from 2006 to 2019.

Implementation of countervailing duties on honey imported only from specific countries requires identification of source countries; attempted evasion by mislabeling is common. Microscopic, manual identification of pollen from plants characteristic of given countries of origin is an established technique, though coverage is not perfect. More importantly, this approach is laborious and time-consuming and relies on a small number of deeply-knowledgeable experts, and fails to identify samples that have under-studied pollens. This project is motivated by the expressed interest of CBP LSS personnel in reliable and scalable methods of identifying the sources of honey samples. The distinguishing power of DNA in natural products is exceptionally large, and DNA analysis technology is rapidly advancing; the cost of DNA sequencing recently has decreased by orders of magnitude. The proposed DNA technology for pollen tracing also could find broad applications in other forensic applications, including identifying the origin and species of other natural products, and tracing the origins of shipments, narcotics, and persons.

The DNA approach facilitates the exploitation of information from new samples, both by clustering with known standards and likely also by exploitation of trace plants known to be limited to particular geographic origins.[2,3] DNA methods also can address the problem of filtered honey source identification since trace DNA remains in filtered honey, as discussed below. While several researchers have demonstrated the potential of pollen DNA to identify the floral source of honey,[4,5] this work is not focused or organized so as to be routinely useful to CBP. There is only early-stage research on PCR identification[6–9]of floral sources of honey, with little focus on geographical attribution. There is almost no research on the DNA content of filtered honey.

CBP's trade enforcement operational approach is based on "DETECTING high-risk activity, DETERRING non-compliance, and DISRUPTING fraudulent behavior."[10] (https://www.cbp.gov/document/fact-sheets/cbp-trade-enforcement-operational-approach), and each of these can be advanced by the proposed work. The project offers: (1) routine, lower-skill, and cost-efficient DNA sequencing-based identification of honeys' countries of origin, (2) reliable sourcing of filtered honey, and (3) fast-turnaround, cheaper DNA-amplification sourcing tools, like those used for diagnosing infections. We confidently predict that DNA identification of honey sources will be as reliable as microscopy-based melissopalynology, and that it will be more easily extended to regions for which few characteristic plant species are currently known. DNA found in honey from a region of interest can readily be added to the custom database we are assembling.

In this work, we have assembled country-specific plant DNA sequences isolated from pollens of 303 honey samples collected across 37 countries, as shown in Figure 3. We explored amplicon-based next-generation sequencing (NGS) of plant gDNA isolated from pollen in honey as described in Figure 2. We targeted the internal transcribed spacer 2 (ITS2) region separating the large subunit genes (5.8S and 28S) of plant genomic DNA (gDNA) for two reasons. First, ITS2 is present in multiple copies in the plant genome, making it an easy target for amplification[11]. Secondly, it is known to facilitate species-level discrimination of plants[12–14].

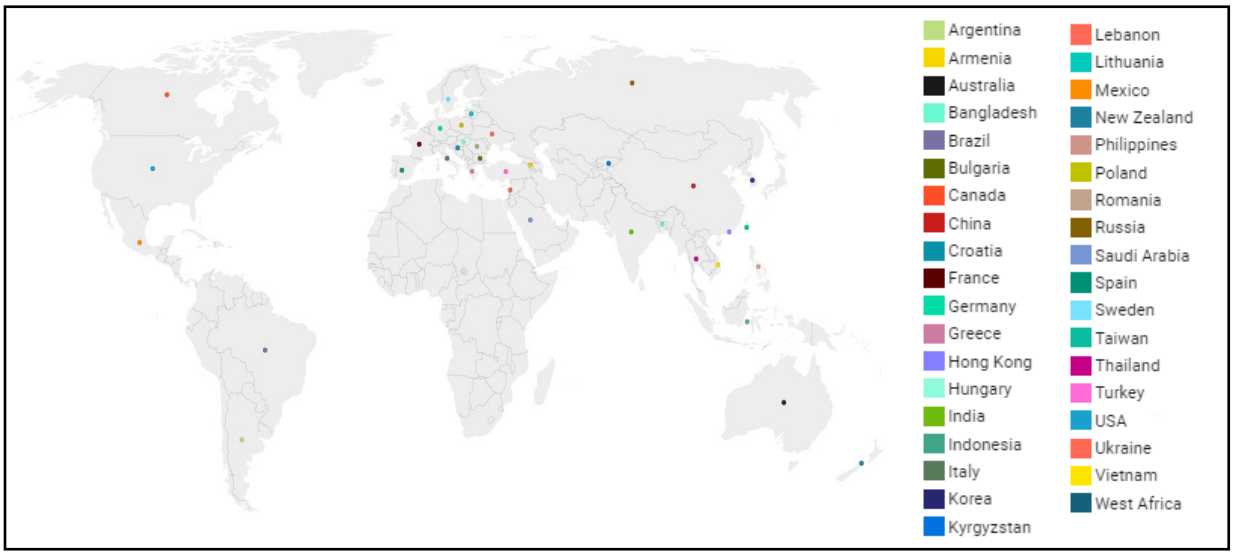


Figure 3. Distribution of 303 honey samples across 37 countries

Finally, in this work we have isolated, PCR-amplified and sequenced the free DNA in filtered honey, a very promising opportunity for honey source identification even after attempted evasion, as shown in Figures 4 and 6.

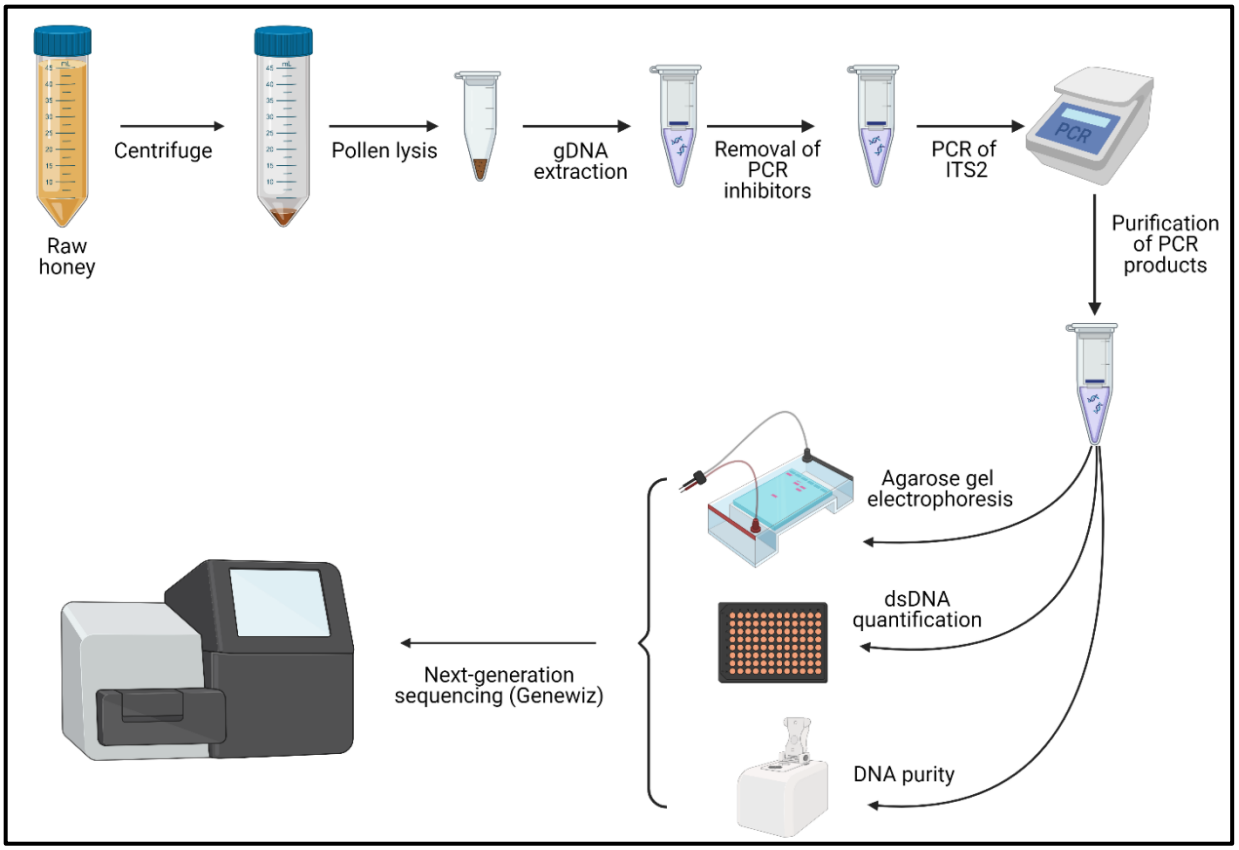


Figure 4. Schematic for pollen DNA sequencing

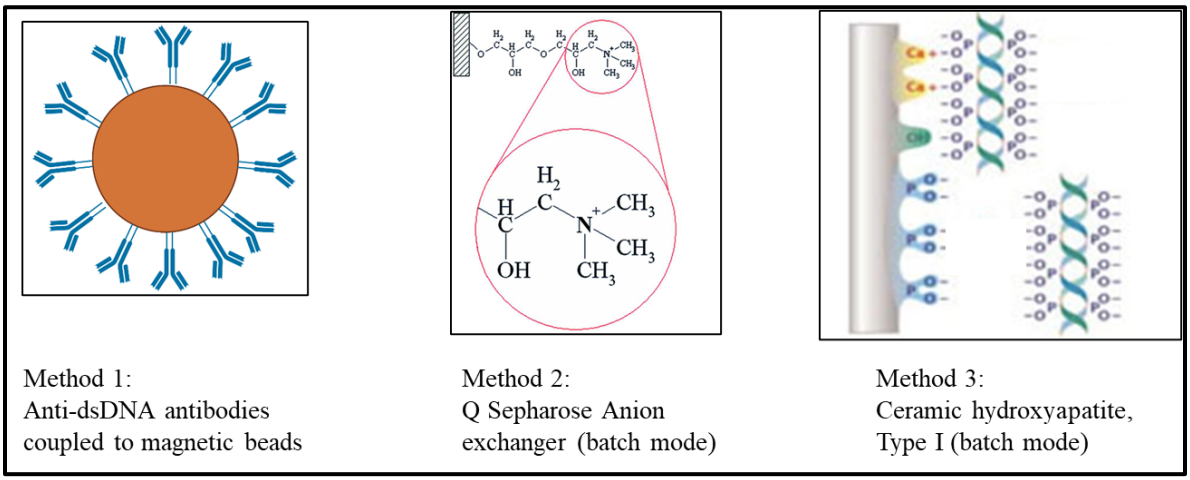


Figure 5. Different methods for capturing soluble DNA

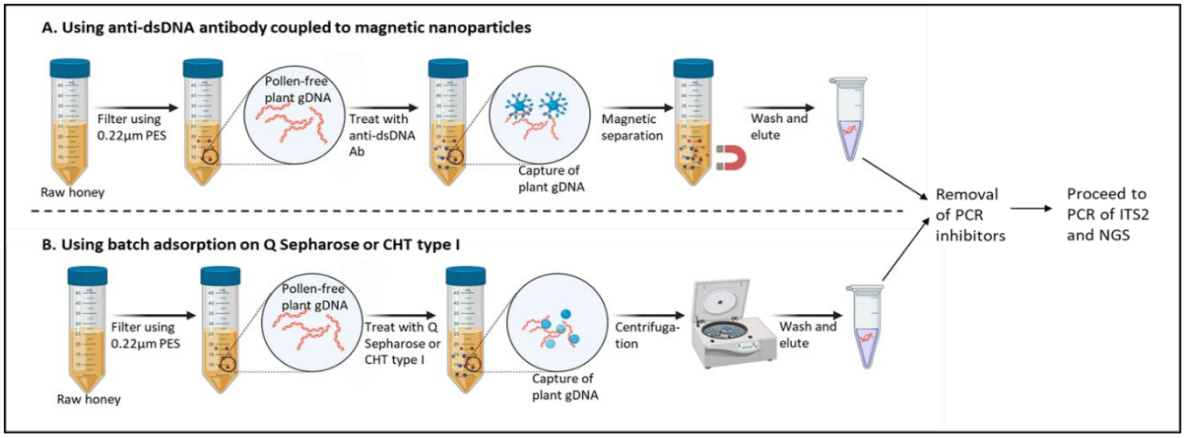


Figure 6. Schematic of capture of plant gDNA from pollen-free filtered honey

2. MATERIALS AND METHODS

2.1. Reagents

Anti-ds DNA antibody (35I9 DNA; ab27156) was purchased from Abcam (Cambridge, Massachusetts). Promag amine beads (3.10 μm, PMA3N) were from Bangs Laboratories, Inc. (Fishers, Indiana). Zeba™ spin desalting columns (40K MWCO, 0.5 mL, 87766), AminoLink™ Reductant sodium cyanoborohydride (44892), and SYBR™ Safe DNA Gel Stain (S33102) were purchased from ThermoFisher Scientific (Carlsbad, California). Amicon Ultra-0.5 centrifugal filters (100 kDa, UFC510096) were from MilliporeSigma (Burlington, Massachusetts). Phosphate-buffered saline (PBS) tablets, pH 7.4 were from Takara Bio USA Inc. (Mountain View, California).

MilliporeSigma™ Steriflip™ Sterile Disposable Vacuum Filter Units (polyethersulfone membrane, 0.22 μm, SCGP00525) and molecular biology grade ethanol (BP2818500) were purchased from Fisher Scientific (Hanover Park, Illinois). CHT™ ceramic hydroxyapatite, Type I (40 μm particle size) was from Bio-Rad (Hercules, California). Q Sepharose® Fast Flow (wet bead size 45-165 μm, preswollen in 20% ethanol, Q1126-100ML), Nuclease-Free Water, for Molecular Biology (W4502), Corning® 96-well Black Flat Bottom Polystyrene NBS Microplate (3991), Trizma® base (T6066), Hydroxylamine hydrochloride (159417), Glycine (G7126), and Ethylenediaminetetraacetic acid disodium salt dihydrate (EDTA, molecular biology grade, E5134) were purchased from Sigma Aldrich (St. Louis, Missouri).

Buffer PB (19066), Buffer PE (concentrate, 100 ml, 19065), DNeasy Plant Mini Kit (69104) QIAquick Spin Columns (28115), QIAquick® PCR Purification Kit (28104), Nuclease-Free Water (129117), and QIAGEN Proteinase K (19131) were purchased from Qiagen (Germantown, Maryland). Q5® Hot Start High-Fidelity 2X Master Mix (M0494S), Gel loading buffer Purple (6X, B7024S), and Proteinase K (Molecular Biology Grade, P8107S) were purchased from New England BioLabs Inc. (Ipswich, Massachusetts).

Plant ITS2 primers used in this study were as previously reported by Chen *et al.*[13] Forward primer (20 nt, 5'- ATG CGA TAC TTG GTG TGA AT -3') and Reverse primer (21 nt, 5'-GAC GCT TCT CCA GAC TAC AAT-3') were purchased from Integrated DNA Technologies, Inc. (Coralville, Iowa). Mx3000P optical strip tubes (401428) and Mx3000P optical strip caps (401425) were from Agilent Technologies, Inc. (Santa Clara, California). Eppendorf DNA LoBind Tubes, 2.0 ml, PCR clean, colorless (4043-1048) were purchased from USA Scientific, Inc. (Orlando,

Florida). Agarose Med EEO (A1035) was from US Biological Life sciences. QuantiFluor® dsDNA System (E2670) was purchased from Promega (Madison, Wisconsin).

2.2. Honey samples

The COVID-19 pandemic prevented planned travel to countries of interest to obtain perfectly-provenanced honey samples. Three hundred and three different raw honey samples purchased from local grocery stores, apiaries, or provisionally-trusted online suppliers or obtained directly from hives were selected for this study (Table 2 gives a list of Manuka samples). Each of the honey samples was processed to obtain plant gDNA from its pollen, as described in section 2.3. Additionally, we analyzed ultra-filtered honey to isolate trace soluble plant gDNA using three methods as described in section 2.4.

2.3. Extraction of plant gDNA from pollen

We developed a method based on that of Soares *et al.*[15] to isolate plant genomic DNA (gDNA) from pollen. Approximately 15 g raw honey sample was weighed in a sterile 50 ml centrifuge tube and its weight was then made up to 50 g using nuclease-free water. The diluted honey sample was then heated for 15 min in a water bath maintained at a constant temperature of 56ºC. This allowed the homogeneous mixing of honey with water. The tubes were centrifuged at 4000 *g* for 30 min, at room temperature. After centrifugation, the supernatant was discarded and the pellet containing pollens was then transferred to a sterile 2 ml microcentrifuge tube. The pellet was washed using 2 ml nuclease-free water and recentrifuged at 4000 *g* for 15 min. The supernatant was discarded, and the pollen pellet was resuspended in 100 µl nuclease-free water.

The separated pollens were pulverized by vortexing for 2 min at high speed with 7-8 sterile glass beads. The disrupted pollens were transferred to a new sterile 2 ml microcentrifuge tube. At this stage, the samples can be stored at -20ºC until the next step. For efficient extraction of plant gDNA, at least 100 mg (wet weight) of pollen pellet was processed for each sample. In samples with lower pollen contents, two separate 15 g honey samples were processed as described above and pooled.

Plant gDNA was extracted using Qiagen's DNeasy Plant Mini Kit. The pollen pellet (100 mg) was treated with 400 µl Buffer AP1 (provided in the kit) and 25 µl of Proteinase K (20 mg/ml; NEB). The pellet was vortexed at medium speed for efficient lysis and to prevent shearing of gDNA. The treated pellet was incubated at 56ºC for 10 min, then allowed to cool for 2 min at room

temperature. To this was added 4 µl RNase A (100 mg/ml; provided in the kit). The tube was mixed gently by inverting 3-4 times and was then incubated at 65ºC for 10 min. The rest of the steps for extraction of plant gDNA followed as per the manufacturer's (Qiagen's) instructions, and finally, gDNA was eluted in 200 µl of Buffer AE (provided in the kit). The introduction of Proteinase K treatment into this modified protocol helped improve plant gDNA extraction success rate, irrespective of the sample type.

The resulting crude extract of plant gDNA was further purified using Qiagen's QIAquick® PCR Purification Kit to remove PCR-inhibitory components. Briefly, 200 µl of plant gDNA present in Buffer AE was mixed with 1000 µl Buffer PB by aspirating 7-8 times using a microtip. The rest of the steps for purification of plant gDNA were as per the manufacturer's instructions, and finally, gDNA was eluted in 50 µl of Buffer EB (10 mM Tris·Cl, pH 8.5; provided in the kit). The extracted and purified plant gDNA was then stored at -20ºC until further use. Repeated freeze-thaw cycles and vortexing of extracted DNA were avoided to prevent degradation of DNA.

2.4. Methods for the capture of trace soluble DNA from filtered honey

We explored three different methods for capturing and enriching the traces of soluble plant gDNA present in ultrafiltered honey samples devoid of pollen.

2.4.1. Batch adsorption on an anion-exchanger

Approximately 15 g of raw honey sample was weighed in a 50 ml sterile centrifuge tube and its weight was then made up to 50 g using 20 mM Tris containing 150 mM NaCl (pH 8.42). The diluted honey sample was heated for 15 min in a water bath maintained at a constant temperature of 56ºC. This allowed the homogeneous mixing of honey with the buffer. The pH of the honey sample was adjusted to 8.5 before filtration (raw honey can have a pH as low as 4.5; pH adjustment was important for efficient capture). The honey sample was then filtered using a sterile disposable vacuum filter unit (PES membrane, 0.22 µm). Alternatively, honey samples with low pollen content were prepared as described above. But, instead of filtering using a 0.22 µm membrane, the samples were centrifuged at 4000 *g* for 30 min to separate out the pollen. The supernatant containing pollen-free plant gDNA was treated with the resin. The filtered honey sample was contacted with an anion exchange adsorbent for the capture of soluble DNA, as described below.

Briefly, Q Sepharose® Fast Flow resin was first uniformly mixed, and 500 µl of resin slurry was then pipetted into a 15 ml sterile centrifuge tube. The resin was washed with 10 ml of 20 mM Tris containing 150 mM NaCl (pH 8.42) by centrifugation at 2000 *g* for 20 min. The supernatant was discarded, and the settled resin was resuspended in 1.5 ml of 20 mM Tris containing 150 mM NaCl (pH 8.42) to obtain 30% resin slurry (v/v). The 30% resin slurry was then added to the 50 ml tube containing the filtered honey sample. The tube was then placed on a rotator for 1 h at room temperature (28 rpm, Model #RT50, Cole-Parmer, Vernon Hills, Illinois).

A portion from a 50 ml tube treated with the resin was transferred to a 15 ml sterile tube and centrifuged at 2000 *g* for 10 min. The supernatant was discarded, and the resin was collected. This step was repeated until all resin was collected in the same 15 ml tube. The resin was washed twice with 5 ml 20 mM Tris containing 400 mM NaCl (pH 8.42) by centrifugation at 2000 *g* for 10 min. Finally, to elute the captured plant gDNA from the resin, 1.5 ml of 2 M NaCl (elution buffer) was added to the washed resin. The resin suspended in the elution buffer was transferred to 2 ml sterile tubes and incubated on a rotator for 30 min at room temperature (28 rpm). The resin was centrifuged at 2000 *g* for 10 min. The resin settled at the bottom of the tube and the supernatant contained the eluted plant gDNA.

The supernatant (approximately 1.5 ml) containing the isolated soluble plant DNA was then transferred without disturbing the resin pellet to a new 15 ml tube. The 1.5 ml supernatant was then mixed with 7.5 ml of Buffer PB by aspirating gently 5-6 times using a sterile 1 ml microtip. This mixture was then concentrated using a DNeasy Mini spin column from Qiagen's DNeasy Plant Mini Kit. All centrifugation steps were performed at centrifugation at 10,000 *g* for 1 min. The column was then washed with 500 µl AW2 and centrifuged at 10,000 *g* for 1 min. A second wash of AW2 was repeated at 20,000 *g* for 2 min. Finally, pollen-free plant gDNA was eluted in 200 µl of Buffer AE and was stored overnight at -20ºC until the next step.

The resulting crude extract of plant gDNA was further purified using Qiagen's QIAquick® PCR Purification Kit to remove PCR-inhibitory components. Briefly, 200 µl of plant gDNA present in Buffer AE was mixed with 1000 µl Buffer PB by aspirating 7-8 times using a microtip. The rest of the steps for purification of plant gDNA were as per the manufacturer's instructions, and finally, gDNA was eluted in 50 µl of Buffer EB (10 mM Tris·Cl, pH 8.5; provided in the kit). We repeated the silica treatment by mixing 50 µl eluted plant gDNA with 250 µl of Buffer PB. The rest of the steps remained the same as described above after buffer PB. Finally, the plant

gDNA was eluted in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5) and stored at -20ºC until the next step. Repeated freeze-thaw cycles and vortexing of extracted DNA were avoided to prevent degradation of DNA.

2.4.2. Batch adsorption on ceramic hydroxyapatite (CHT) type I

Approximately 15 g of raw honey sample was weighed in a sterile 50 ml centrifuge tube and its weight was made to 50 g using 10 mM $NaPO_4$ containing 1 mM EDTA and 0.5 M NaCl (pH 7.0). The diluted honey sample was heated for 15 min in a water bath maintained at a constant temperature of 56ºC. This allowed the homogeneous mixing of honey with the buffer. The pH of the honey sample was adjusted to 7.5 before filtration. The honey sample was then filtered using a sterile disposable vacuum filter unit (PES membrane, 0.22 µm) to remove all pollens. The filtered honey sample was treated with CHT for the capture of soluble DNA as described below.

Briefly, 500 mg CHT adsorbent was first weighed into a sterile 15 ml centrifuge tube. The adsorbent was washed with 10 ml of 10 mM $NaPO_4$ containing 1 mM EDTA (pH 7.0) by centrifugation at 750 g for 5 min. The supernatant was discarded, and the settled adsorbent was then resuspended in 1.5 ml of 10 mM $NaPO_4$ containing 1 mM EDTA (pH 7.0) to obtain 30% adsorbent slurry (w/v). The 30% adsorbent slurry was then added to the 50 ml tube containing the filtered honey sample. The tube was then kept on a rotator for 1 h at room temperature (28 rpm).

The sample from the 50 ml tube treated with the adsorbent was transferred to a 15 ml sterile tube and centrifuged at 750 *g* for 2 min. The supernatant was discarded, and the adsorbent was collected. This step was repeated until all adsorbent material was collected in the same 15 ml tube. The adsorbent was washed twice with 5 ml of 10 mM $NaPO_4$ containing 1 mM EDTA (pH 7.0) by centrifugation at 750 *g* for 2 min. Finally, to elute the captured plant gDNA from the resin, 400 mM $NaPO_4$ containing 1 mM EDTA (pH 7.0) was added to the washed adsorbent. The adsorbent suspended in the elution buffer was transferred to a 2 ml sterile tube and incubated on a rotator for 30 min at room temperature (28 rpm). The tube was centrifuged at 750 *g* for 2 min. The adsorbent settled at the bottom of the tube, and the supernatant contained the eluted plant gDNA.

The supernatant (approximately 1.5 ml) containing the isolated soluble plant gDNA was then transferred without disturbing the pellet to a new 15 ml tube. The 1.5 ml supernatant was then mixed with 7.5 ml of Buffer PB by aspirating gently 5-6 times using a sterile 1 ml microtip. This mixture was then passed by centrifugation through a Qiagen QIAquick® Spin Column (silica mini-

column) to promote the binding of DNA. All centrifugation steps were performed at 17,000 *g* for 1 min. The plant gDNA bound to the silica column was then washed using 750 µl Buffer PE by centrifugation. The column was centrifuged again to remove any traces of Buffer PE. The plant gDNA was eluted in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5). We repeated the silica treatment by mixing 50 µl eluted plant gDNA with 250 µl of Buffer PB. The rest of the steps were as described above after buffer PB. Finally, the plant gDNA was eluted in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5) and stored at -20ºC until the next step.

2.4.3. Using anti-dsDNA antibodies coupled to magnetic microspheres

Anti-dsDNA antibodies coupled to amine magnetic microspheres were prepared using periodate-based carbohydrate oxidation as described in our previous publication.[16] Briefly, 35 µl of anti-dsDNA antibody stock was transferred into 100 mM sodium acetate buffer (pH 5.4) using a Zeba column (40K, 0.5 ml). Fifty µl of the antibody preparation (90 µg/ml) thus obtained was then mixed with 5 µl of 0.1 M $NaIO_4$. The tube containing this mixture (covered with aluminum foil for protection from light) was incubated on a rotator (28 rpm) for 30 min at room temperature. The aldehyde-activated antibodies were then purified and concentrated using a 100 kDa Amicon Ultra centrifugal filter in 200 mM sodium carbonate buffer (pH 9.6). The recovered antibody stock was diluted to 100 µl at 50 µg/ml in 200 mM sodium carbonate buffer (pH 9.6) and was kept on ice until the next step of conjugation.

In another tube 100 µl Promag amine microspheres (3.1 µm; 1×10^8 particles) were washed three times and resuspended in 100 µl of 200 mM sodium carbonate buffer (pH 9.6). The 100 µl washed particles were then mixed with 100 µl of oxidized antibody preparation and incubated on a rotator for 2 h at room temperature. After incubation, 5 µl of 5 M $NaCNBH_3$ made up in 1 M NaOH was added to the reaction and incubated on a rotator for 30 min at room temperature. Unreacted aldehydes were then quenched by adding 75 µl of 1 M hydroxylamine, and the mixture was incubated on a rotator for 30 min at room temperature. The antibody-functionalized magnetic particles were separated by magnetic separation and washed three times using phosphate-buffered saline (PBS; pH 7.4). Finally, anti-dsDNA antibodies coupled to magnetic particles were resuspended in 100 µl PBS (pH 7.4) and stored at 4ºC until further use.

Approximately 15 g of raw honey was weighed in a 50 ml sterile centrifuge tube and its weight was made to 50 g with 25 mM Tris (pH 8.42). The diluted honey sample was heated for 15

min in a water bath maintained at a constant temperature of 56ºC. The pH of the honey sample was adjusted to 8.5, and it was then filtered using a sterile disposable vacuum filter unit (PES membrane, 0.22 µm). The filtered honey sample was treated with anti-dsDNA coupled magnetic particles for the capture of soluble DNA, as described below.

Briefly, 20 µl anti-dsDNA coupled magnetic particles were washed three times with 25 mM Tris (pH 8.42) and resuspended in 20 µl 25 mM Tris (pH 8.42). The particles were then added to the 50 ml tube containing the filtered honey sample. The tube was then kept on a rotator for 1 h at room temperature (28 rpm). The sample from the 50 ml tube treated with the particles was then concentrated in a 2 ml tube using magnetic separation until all particles from the 50 ml tube were collected.

The particles were washed twice with 5 ml of 25 mM Tris (pH 8.42) by magnetic separation. Finally, to elute the captured plant gDNA from the anti-dsDNA antibody-coupled magnetic particles, 50 µl of 100 mM glycine (pH 3) was added to the tube. The 50 µl of 100 mM glycine (pH 3) containing eluted plant gDNA was immediately transferred to another 2 ml tube containing 5 µl of 2 M Tris (pH 8.42).

The 55 µl of eluted plant gDNA was then mixed with 250 µl of Buffer PB by aspirating gently 5-6 times using a sterile 1 ml microtip. This mixture was then passed by centrifugation through a Qiagen QIAquick® Spin Column (silica mini-column). All centrifugation steps were performed at 17,000 *g* for 1 min. The plant gDNA bound to the silica column was then washed by centrifugation using 750 µl Buffer PE. The column was centrifuged again to remove any traces of Buffer PE. The plant gDNA was eluted in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5). We repeated the silica treatment by mixing 50 µl eluted plant gDNA with 250 µl of Buffer PB. The rest of the steps were as described above after buffer PB. Finally, the plant gDNA was eluted in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5) and stored at -20ºC until the next step.

2.5. PCR amplification of ITS2

The polymerase chain reaction (PCR) amplification was carried out in a total reaction volume of 50 µl containing 25 µl Q5® High-Fidelity 2X Master Mix, 2.5 µl of 10 µM of each primer, 2 µl of DNA template (approximately 10-50 ng for most pollen samples; in future work this could be standardized) or 10 µl of DNA template (for soluble DNA samples) and nuclease-free water to obtain a final volume of 50 µl. PCR was performed in an MJ Mini thermal cycler

(Bio-Rad Laboratories, Hercules, California) using the following program: (i) initial denaturation at 98ºC for 30 sec; (ii) 40 cycles of 98ºC for 10 sec, 62ºC for 30 sec and 72ºC for 30 sec; and (iii) final extension at 72ºC for 2 min. Modified conditions were used for some samples that failed amplification using the above program. The modified program included: (i) initial denaturation at 98ºC for 30 sec; (ii) 40 cycles of 98ºC for 10 sec, 62ºC for 30 sec and 72ºC for 1 min; and (iii) final extension at 72ºC for 5 min.

The amplified products of plant ITS2 were then purified using a Qiagen QIAquick® PCR Purification Kit as per the manufacturer's instructions. Finally, ITS2 PCR products were eluted in 50 µl of Buffer EB (10 mM Tris·Cl, pH 8.5; provided in the kit). The purified PCR products were then stored at -20ºC until further use. The PCR products were then analyzed in a 1.5% agarose gel electrophoresis and stained using SYBR™ safe DNA gel stain to examine the size of the DNA products obtained. The purified PCR product was also analyzed on a Nanodrop system to study DNA purity by absorbance ratios. The concentration of purified PCR products was determined using the QuantiFluor® dsDNA System.

The purified PCR products had to meet the following criteria to be sent for amplicon-based NGS analysis: i) concentration of products normalized to 20 ng/µl; ii) at least 500 ng of DNA required, and iii) DNA purity index (A260/A280) between 1.8 and 2.0. For each sample of pollen DNA and soluble DNA, two reactions (each of 50 µl) were performed and pooled together to meet the criteria required for NGS analysis. The samples were then sent to Genewiz for Amplicon-EZ analysis (2 x 250 bp sequencing).

2.6. Sequence data analysis

The raw FASTQ files received from Genewiz were analyzed using a bioinformatics pipeline as shown in Figure 5 and adapted from the DADA2 ITS Pipeline Workflow and the workflow for Microbiome Data Analysis.[17,18] The details of the analysis pipeline are explained in Figure 7.

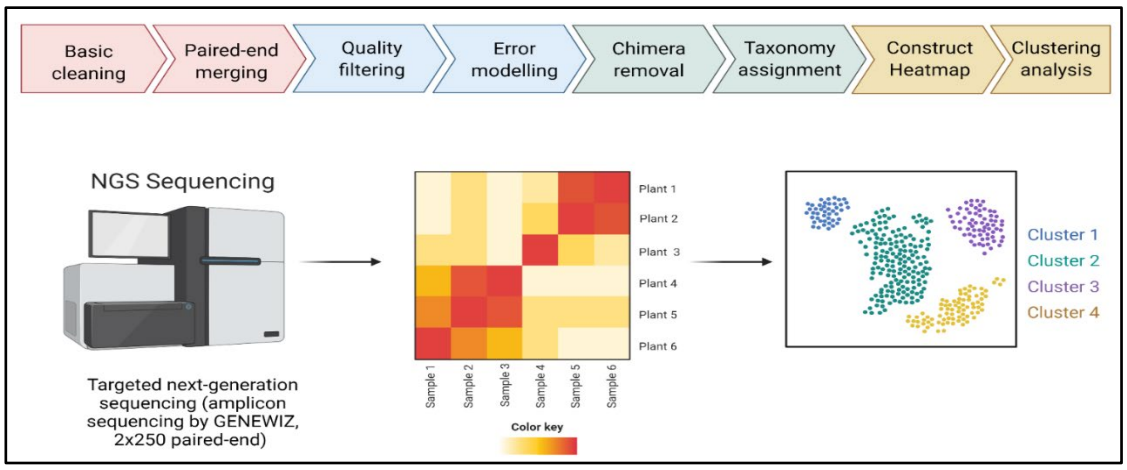


Figure 7. Schematic of bioinformatics workflow for plant ITS2 from honey

All raw FASTQ files received from Genewiz were random mixtures of forward and reverse reads instead of the typical case where R1_001 and R2_001 files contained forward and reverse reads respectively (confirmed by correspondence with Genewiz). The first step of the analysis was to segregate forward and reverse reads into R1_001 and R2_001 files, respectively. The first 20 or 21 bases can be used to identify whether a read is forward or reverse based on its sequence matching with the primer sequences. However, sequencing errors can introduce mismatches into otherwise valid reads. On the other hand, excessively relaxing the stringency in sequence matching will lead to misidentification of the reads. To balance this the paired-end reads that had 5 nt or less primer mismatches were retained. A cutoff of 5 for the maximum number of mismatches was selected to retain the maximum number of unique reads mapped to the forward and reverse primers.

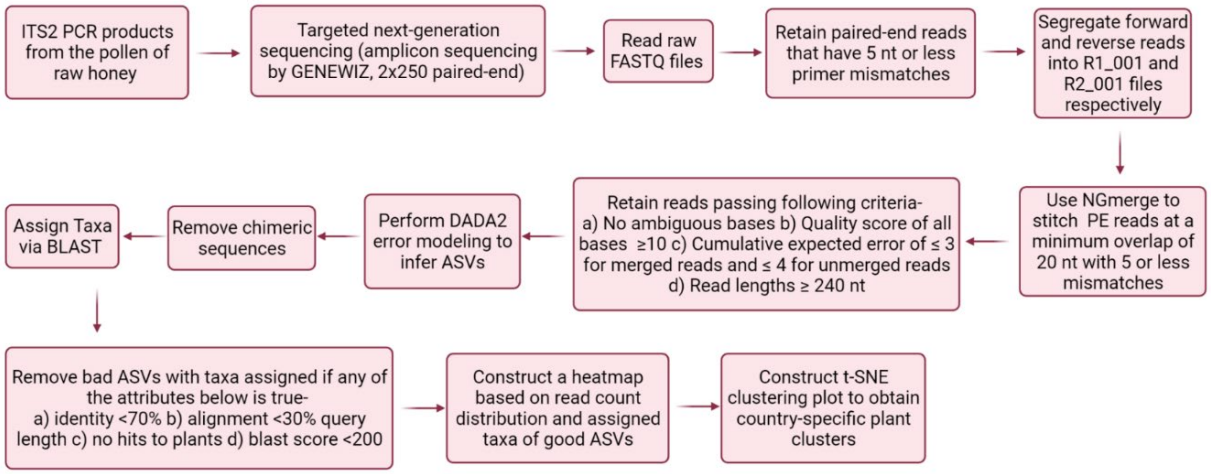


Figure 8. Bioinformatics steps

The next step was to stitch the paired-end reads using NGmerge[19] at a minimum overlap of 20 nt, allowing for a maximum of 5 nt or fewer mismatches. Thus, after merging, we had some reads that contained both forward and reverse primers and were considered full reads (or merged reads) and spanned the entire ITS2 sequence. But some ITS2 amplified regions exceeded the maximum effective read length of the sequencing platform, which is 480 nt (2×250 read length – 20 nt minimum overlap) because the (conserved) priming sites are set back from the (variable) ITS2 sequence. The paired-end reads with ambiguous bases were removed and reads that successfully merged and forward reads that failed to merge were further processed.

NGS reads were filtered further to facilitate DADA2[20] error modeling to amplicon sequence variants (ASVs). All reads with ambiguous bases (Ns), reads with bases with quality scores below 10, and all reads with a cumulative expected error greater than 3 for merged reads or greater than 4 for forward reads were filtered. The filtering parameters were chosen as such to refine error modeling for the DADA2 algorithm downstream of the analysis by improving read quality while retaining as many reads as possible. The resulting ASVs were further analyzed to remove any unusually short (<240 nt) and chimeric sequences that were present.

ASVs were searched against NCBI's nucleotide database (blastn) to identify the source organism. Results were restricted to the top ten sequences producing significant alignments and limited to records that include Viridiplantae (taxid:33090).

Each ASV is assigned a species based on the top blast hit. ASVs that are short or repetitive usually don't have a good blast hit in the GenBank database. We used this fact to further filter ASVs. ASVs with top blast hits that satisfy at least one of the criteria below were filtered out for low quality:

a) percent identity in the alignment less than 85
b) alignment length less than 1/3 of the length of the ASV (query)
c) alignment is less than 150 bp long
d) blast bit score is less than 200
e) species name includes a match to "environmental sample" or matches 'N/A'

After filtering out low-quality ASVs, we also filtered out samples with low complexity. For each sample, we counted the number of ASVs detected in the sample with at least 10 reads. If the sample had less than 3 ASVs that satisfied these criteria it was termed a 'low complexity sample' and removed from the analysis.

We also tag 'species-specific' blast hits. In cases where there are multiple plant species in blast hits, but the best blast score is unique to a species, and all the other species have lower blast scores, we tag these blast hits as 'species-specific'. We used the Kew Royal Botanic Gardens Plant of the World online database to determine the range of each detected species (www.plantsoftheworldonline.org).

Heatmaps and t-SNE clustering plots were constructed to analyze the abundance of different plants across all honey samples. We used tSNE (T-Distributed Stochastic Neighbor Embedding) method as implemented in the Rtsne R package for clustering of data. t-SNE is an unsupervised non-linear dimensionality reduction and data visualization technique that takes high-dimensional data and reduces it to a low-dimensional graph. It is similar to the well-known PCA (Principal component analysis) method but unlike PCA, t-SNE can reduce dimensions with non-linear relationships. Read counts were expressed as fractions of the total number of reads per sample.

3. RESULTS AND DISCUSSION

3.1. Amplification of ITS2

We were able to successfully isolate and amplify plant ITS2 from honey samples using a primer pair published by Chen *et al.*[13] The ITS2 region in plants usually varies from ~180-390 bp.[21,22] The forward and reverse primers anneal in the conserved regions of 5.8S (~85 bp upstream of ITS2) and 26S (~142 bp downstream of ITS2) of plant gDNA respectively. In accordance with these published results, we observed varying lengths of ITS2 amplicons in our honey samples ranging from 100 bp to 700 bp as shown in Figures 7 and 8. For each of these samples, 10 µl amplified ITS2 product was mixed with 2 µl of gel loading buffer to analyze on 1.5% agarose gel and was finally post-stained with SYBR™ safe DNA gel stain for 30 min. We observed smearing in a significant fraction of the amplified products as seen in Lane 11 of Figure 9. This smearing of samples was mainly associated with the poor quality of isolated plant gDNA from pollen by the standard protocol. Additionally, we speculate some of the smearing might also be due to fragmentation of plant gDNA during packaging or long-term storage of honey before we acquired and processed it. Under lockdowns and pandemic conditions, we were not able to address this problem in the short term of the project, but we speculate that it could be resolved with further optimization/scaling of DNA isolation procedures, iterative purification, and standardization of DNA concentrations.

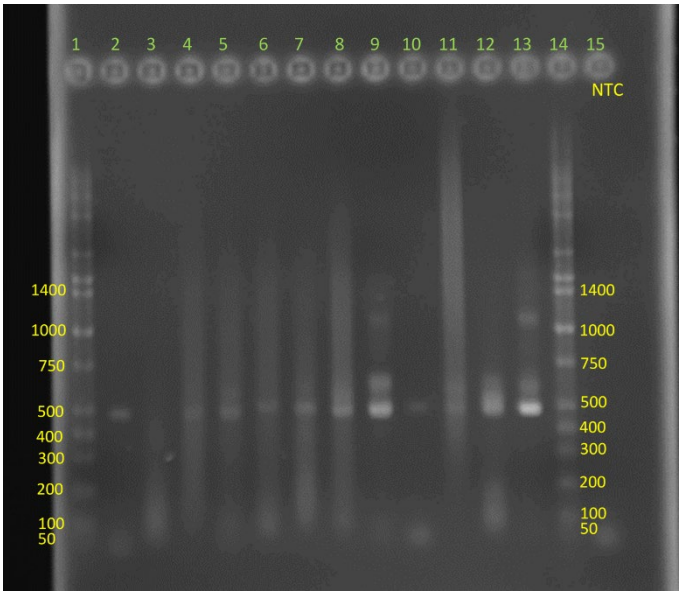


Figure 9. Agarose gel electrophoresis of ITS2 products from raw honey (gel 1).

Lanes 1 and 14: Hi-Lo DNA ladder, Lane 2: Madhava (Brazil and Mexico), Lane 3: Aires del campo (Mexico), Lane 4: La Melipona (Mexico), Lane 5: Sábila y miel (Tsitsilche flowers; Mexico), Lane 6: Gradina multiflower honey (Bulgaria), Lane 7: Livada love nature (Linden honey; Romania), Lane 8: Hilltop honey (Scottish Heather honey; UK), Lane 9: Mieli Thun (Forest honeydew honey; Italy), Lane 10: Donoxti (Multiflora blossom honey; Mexico), Lane 11: Brietsamer golden selection raw honey (Germany), Lane 12: R C. Stevenson & Father (Pure Ontoria honey; Canada), Lane 13: 9th meadow honey (Canada) and Lane 15: No-template control.

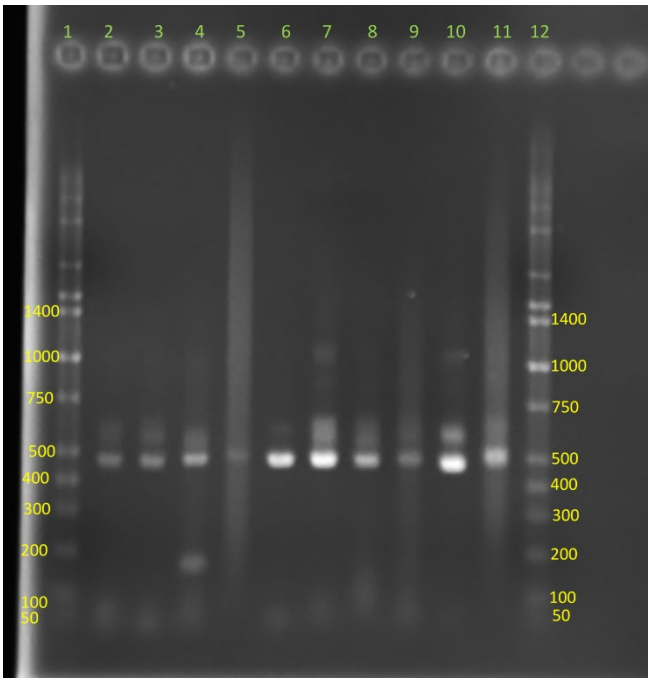


Figure 10. Agarose gel electrophoresis of ITS2 products from raw honey (gel 2).

Lanes 1 and 12: Hi-Lo DNA ladder, Lane 2: Altay mountain honey (Russia), Lane 3: Breitsamer Honig (Blossom honey; Germany), Lane 4: Ceimaya (Mesquite honey; Mexico), Lane 5: Aires del campo (Dzidzilche flowers; Mexico), Lane 6: Bee harmony (Eucalyptus honey; Brazil), Lane 7: Vila vella (Eucalyptus honey; Spain), Lane 8: Langanese forest honey (Germany), Lane 9: Sleeping bear farm (star thistle honey; USA), Lane 10: Attiki Pure Greek honey (Greece), and Lane 11: Ancient Foods (Thyme; Greece).

3.2. Pollen DNA Sequencing

Most of our ITS2 amplified products had concentrations between 40 ng/µl and 120 ng/µl, A260/280 ratio between 1.8 and 2.0, and A260/230 ratio between 2.2 and 2.4. Each of the samples

was normalized to 20 ng/µl before sending to Genewiz for NGS (Amplicon-EZ analysis; 2 x 250 bp sequencing).

As discussed below, some samples did not sequence well. Read numbers for successful samples varied between ~5,000 and 210,000 raw reads per sample. A few samples gave raw read numbers as low as 5,000, which we believe is due to the difficulty of amplifying small amounts of complex plant gDNA target isolated from a complex matrix. Using the rather strict filtering parameters we established for our pipeline, we retained at the end 20-94 percent of total reads. After processing through DADA2, we obtained 9645 ASVs from 303 honey samples. This number decreased to 7613 ASVs after filtering for chimeric sequences.

Additional strict filtering based on blast hit results further eliminated 12% of ASVs (900/7613). After filtering out low-quality ASVs our sample filtering strategy flagged 51% of the samples (175/343; the 343 samples include pollen samples, pollen DNA replicates, and soluble DNA trial experiments) as having low complexity. These blast result-based criteria set for filtering both ASVs and samples are quite strict, but for the first pass at clustering of this large dataset, we wanted to include only data of the highest quality.

3.2.1. t-SNE clustering results

We used the tSNE method for clustering of read count and ASV data. On the clustering plots, the closeness of points on the plot indicates the similarity of results based on ASV relative counts as shown in Figure 11.

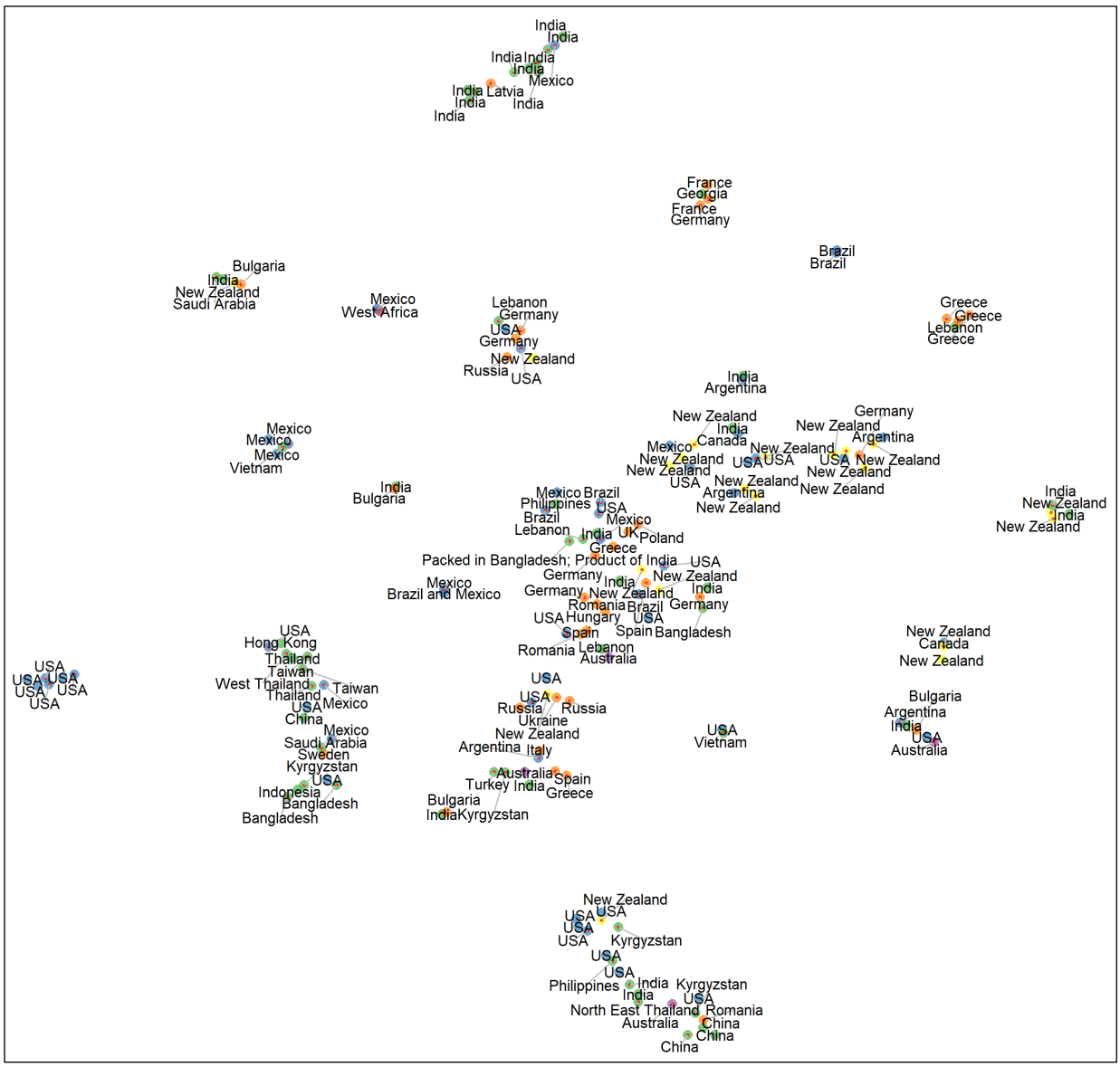


Figure 11. t-SNE plot to study the relationships between 303 honey samples

Next, we evaluated the clusters created by the tSNE algorithm in order to improve the filtering and clustering strategy. Using a custom Perl script, we reviewed the top ASVs, the number of underlying reads and the associated blast hits for each sample in order to review the underlying data for each cluster. The initial analysis mostly produced clusters that were not very region- or country-specific. We analyzed the underlying data to understand why this was happening.

3.2.2. Cosmopolitan plant species

Some commercially-important plants are very widely distributed and skew clustering. For example, we discovered a cluster of samples (shown in Figure 12) that is mostly driven by signals from two species: sunflower and watermelon. These samples come from widely-separated countries and most likely result from the common practice of placing honeybee hives near agricultural fields. Sunflower is one of the top 3 most abundant plants (plants with the highest number of reads) in 30 samples, and watermelon in 16 samples in our dataset. Removing species that are common in many regions of the world could improve clustering. Alternatively, sequencing of additional barcoding regions and/or long-read sequencing could enable finer differentiation among sunflower varieties, or even finer distinctions.

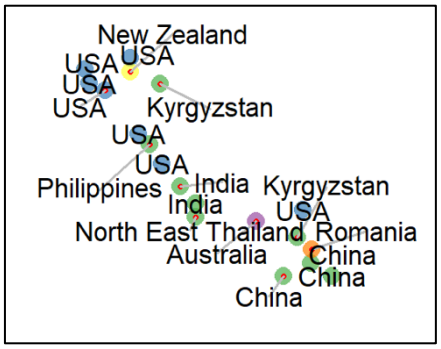


Figure 12. False clustering of samples between non-geographically-related regions of the world

3.2.3. Region-specific clusters

We identified a cluster of honey samples from Texas, dominated by pollen from the oak tree (*Quercus rubra*) and crepe myrtle (*Lagerstroemia indica*) (samples H9, H10, H11, H12, H13, and H15). Five of the samples in this cluster are from a Houston beehive, collected at different areas of the hive, representing various times in the year. The beehive is located on a residential street lined with oak trees and several crepe myrtle trees. The crepe myrtle is actively pollinated by bees, but oaks are wind-pollinated, and therefore the pollen is most likely passively incorporated into the honey.

We identified a cluster of 4 samples (H76, H60, H75, and H251): 3 honey samples from Greece, and one sample from Lebanon as shown in Figure 13 alongside. Species *Erica manipuliflora* was the most abundant species in 3 samples, and in the top 3 hits for the fourth. This species has a narrow range near the Mediterranean Sea: Albania, Cyprus, Greece, Italy, Kriti, Lebanon, Syria, Sicily, Turkey, and former Yugoslavia. We also identified

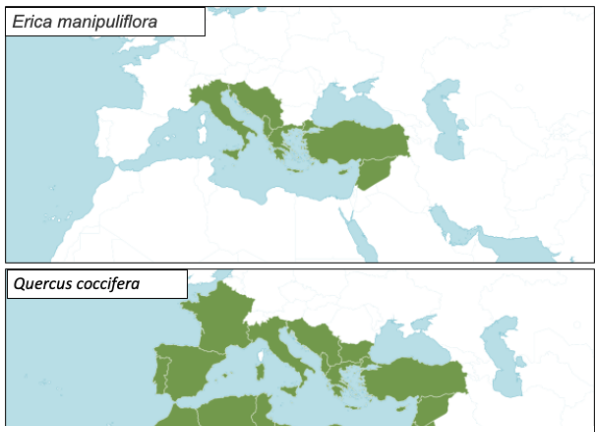


Figure 13. European region-specific cluster of plants.

in two samples from Greece *Quercus coccifera*, which also has a narrow range in the Mediterranean region.

We identified a cluster of 4 samples (Figure 14) from Europe and the country of Georgia (H232, H226, H242, H231; France, Georgia, Germany, France) with several top hits to chestnut, with two of the samples labeled as chestnut honey.

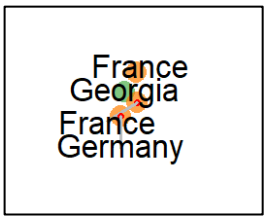


Figure 14. European region-specific cluster 2 of plants.

We also identified a cluster of 4 samples: 3 samples of honey from Mexico (H44, H52, and H43) and one from Vietnam (H263) as shown in Figure 15 alongside. In the sample from Vietnam, we identified 2 species (*Prosopis glandulosa*, and *Viguiera seemannii*), with a native range in central and South America. This fact combined with close clustering of these samples with three samples from Mexico puts into question the Vietnamese origin of this honey. This is true in several other samples, where the range of species that we identified from the sequencing data does not match the country of origin on the label.

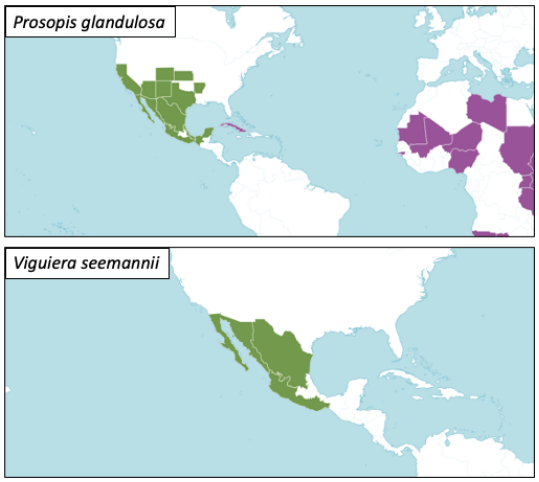


Figure 15. Samples showing DNA sequences originating from different country of origin.

Another example is sample H72 (Breitsamer Honig Blossom honey) is labeled as a product of Germany. But if we look at the top 5 species that are specific by blast (as defined above), four species (*Cissus striata*, *Schinus mole*, *Lithrea ternifolia (*syn *L. molleoides)*, *Amomyrtus luma*) have a native range restricted to South America, and one (*Populus nigra*) has a worldwide cosmopolitan range (Figure 16). It is possible that these samples, where the range of the species we identify from the sequencing data does not match the country of origin on the label, are mislabeled or are a product of mixing honey samples from multiple sources.

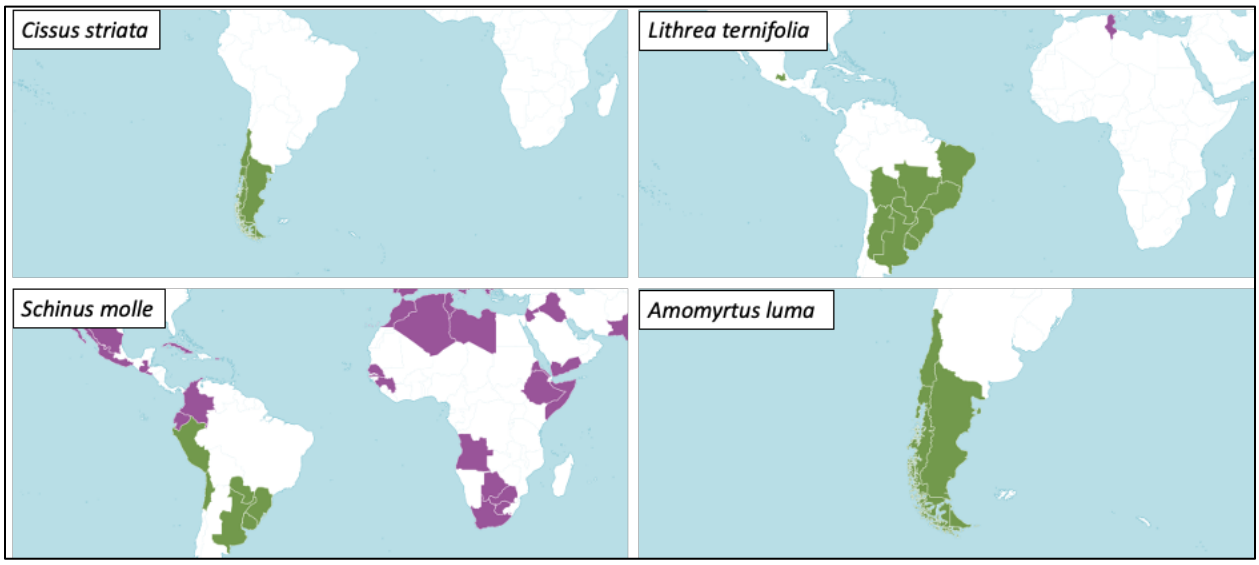


Figure 16. Possible mislabeling of honey samples or blending of honey from different origins of country.

3.2.4. Specific plant origin honey samples

Claims of specific plant origin often are listed on the labels of honey samples, in addition to the country of origin. We aimed to identify how often we see those claimed species in the sample, and at what fraction of reads are they detected in the sample. We had 3 samples of Linden honey (Romania, USA, Bulgaria). Only sample H66 from Romania had detectable reads from linden (*Tilia sp.*), at a very high fraction of 34% (2465/7187). In the other two samples, we did not detect any reads from linden. In sample H67 labeled as Hilltop honey (Scottish Heather honey) from the UK, we detected *Calluna vulgaris*, common heather, at 16% (2584/16111). This species is known to grow widely in Scotland, is native to Europe, Iceland, and the Faroe Islands, and has been introduced into many other places worldwide with suitable climates.

We had 3 samples labeled as Eucalyptus honey (H30, H53, H54), from Australia, Brazil, and Spain. Only two samples (from Spain and Brazil) had reads identified as *Eucalyptus* species. The sample from Brazil had 0.5% (414/78775), while the sample from Spain had 25% (10347/41331) of the reads identified as from *Eucalyptus sp*. Interestingly the sample

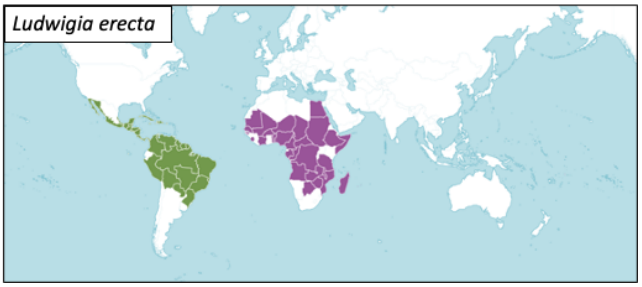


Figure 17. Eucalyptus honey from Australia showing presence of other plant species

from Australia had no reads identified as from Eucalyptus. The most abundant species in that sample was sunflower, and the second most abundant was the south American species *Ludwigia erecta* (Figure 17), raising questions as to the origin of this honey.

One plant species of particular interest is manuka, as manuka honey can fetch an extremely high price, sometimes more than 100-fold higher than other honey types (Figure 18). We collected 37 honey samples labeled as 'manuka honey', of which 21 passed our strict sample filtering criteria. We detected reads matching manuka (*Leptospermum scoparium*) in nine out of the 21 samples, as a varying fraction of total reads (Table 1). The highest count of manuka reads was detected in sample H89 (New Zealand Honey Co., New Zealand), with 4.26% (379/8884) of reads originating from manuka. Two additional samples had about 2% reads originating from manuka; sample H88, Steens Manuka honey from New Zealand (2.03%), and sample H86, Wedderspoon honey from New Zealand (1.9%). One sample had 12 reads (0.29%) detected from manuka (H105), and the other 5 samples had 3 or fewer reads with an average of 0.17% manuka reads overall. Two additional samples contained reads matching kanuka (*Kunzea ericoides*), a species closely related to manuka and also endemic to New Zealand.

Figure 18. Price of Manuka honey in comparison to local grocery store honey.

Table 1. Total reads vs reads of manuka plants in 9 different manuka samples

Sample ID	Manuka reads	Total reads	% Manuka reads
H89	379	8,884	4.27
H88	3	148	2.03
H86	469	24,329	1.93
H90	3	852	0.35
H83	3	1,033	0.29
H105	12	4,175	0.29
H87	2	6,542	0.03
H150	1	4,175	0.02
H85	1	9,598	0.01

3.2.5. Region-specific plants

For our region-specific analysis, we limited the analysis to species-specific blast hits, and required at least 2 reads to be present per ASV to limit spurious results. Our goal was to identify plant species that would be uniquely present in only a specific region or country. In our stringently-filtered set of good samples, we have 33 samples from Europe out of a total of 168 samples.

We first limited our analysis to species that appear only in European samples, and in no non-European samples. We identified one species that was present in four European samples, and three species that were present in three European samples each, and no non-European samples. *Cistus laurifolius*, with a reported range in France, Greece, Italy, Portugal, Spain, and Turkey, was identified in four European samples: Germany, Ukraine, and two samples from Spain. *Helminthotheca echioides* with wide native distribution in Europe and North Africa but also introduced in many other countries, was present in samples from Italy, Greece, and Spain. *Quercus aucheri* native to Turkey was identified in honey samples for Italy, Greece, and Spain. *Erica arborea* with a range in southern Europe, Africa, and the Middle East was identified in one sample from Greece and two samples from Spain.

We postulated that some samples might be of mixed origin, or the country of origin might not be accurately stated on the label. We wanted to identify species present in several European samples, but also allow the presence of the same species in some nominally non-European samples. Therefore, we relaxed our criteria from zero non-European samples and allowed the species to be present in 4 (4/135) or fewer non-European countries, and required it to be present in at least 7 (7/33) of the European countries. This is a 7-fold enrichment (2.9% vs 21%), and a significant difference (p-value = 0.0012, Fisher's exact test). We identified two such specific species. *Quercus robur* with a range in Europe and the Middle East was identified in 7 European samples (Spain, Germany (2), Ukraine, Italy, Greece, and Romania), and 4 non-European samples (USA(2), New Zealand, Canada).

Quercus ilex was identified in 10 European honey samples (Spain (2), Greece (2), Poland, Germany, Romania, Ukraine, Italy, France), and only one non-European sample (the

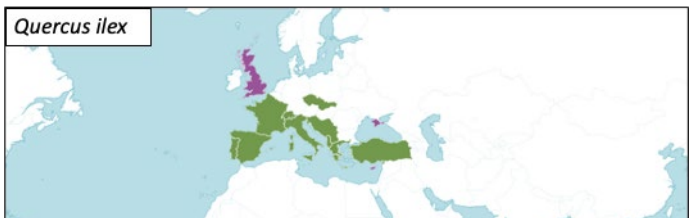


Figure 19. Region-specific plant observed in the European sample.

country of Georgia). Georgia is geographically close to Europe and therefore its flora might overlap those of the neighboring European countries.

We are interested in expanding this analysis for samples from China but are currently limited to only four good samples from China because we could not travel there during the pandemic. We also suspect that some samples that are not labeled as honey from China may be mislabeled (due to widespread mislabeling of honey from China), and this would likely complicate and lower the power of this analysis. We envision that in future stages of this project we could build a model that would integrate information from multiple plant species that even though each might not be exclusive to China, but is enriched in samples from China. Alternatively, once travel is restored, we could create our models based on a highly curated and reliable set of standards where each sample would have a reliably accurate country of origin.

3.3. Soluble DNA Sequencing

We demonstrated our methods for capturing the trace soluble plant gDNA by processing raw honey to establish the relationship between the plant species sequenced from the pollen content of honey and plant species sequenced from the soluble content of the honey. For method development we Kelley's Texas honey, natural raw and unfiltered, USA (Figure 20) purchased from a local grocery store in Houston. We processed a 15 g honey sample for both pollen and soluble plant gDNA to allow direct comparison between pollen DNA and soluble DNA. We studied each of the methods developed in triplicate to address the following topics:

Figure 20. Soluble DNA honey sample

a) Reproducibility of the method when the sample is processed (in terms of yield and quality of the product obtained)
b) Diversity of plant species observed each time we collect a sample for analysis
c) Establishing the relationship between plant species observed and their distribution across the world

The pollen and soluble DNA were processed as described in sections 2.3 and 2.4. The amplified ITS2 products obtained from pollen and soluble DNA were quantified and analyzed by agarose gel electrophoresis. We saw a similar pattern of ITS2 PCR products between pollen DNA and soluble DNA captured by Q Sepharose, as shown in Figure 21. The yield of ITS2 PCR products obtained was also similar for pollen DNA (106 ng/µl) and soluble DNA isolated by Q Sepharose (99.6 ng/µl). Soluble DNA captured by anti-dsDNA Ab coupled to magnetic particles and by ceramic hydroxyapatite (CHT type I) also both gave output but the yield was much less (approximately 25 ng/µl). The low concentrations of the products made them difficult to observe on the gel.

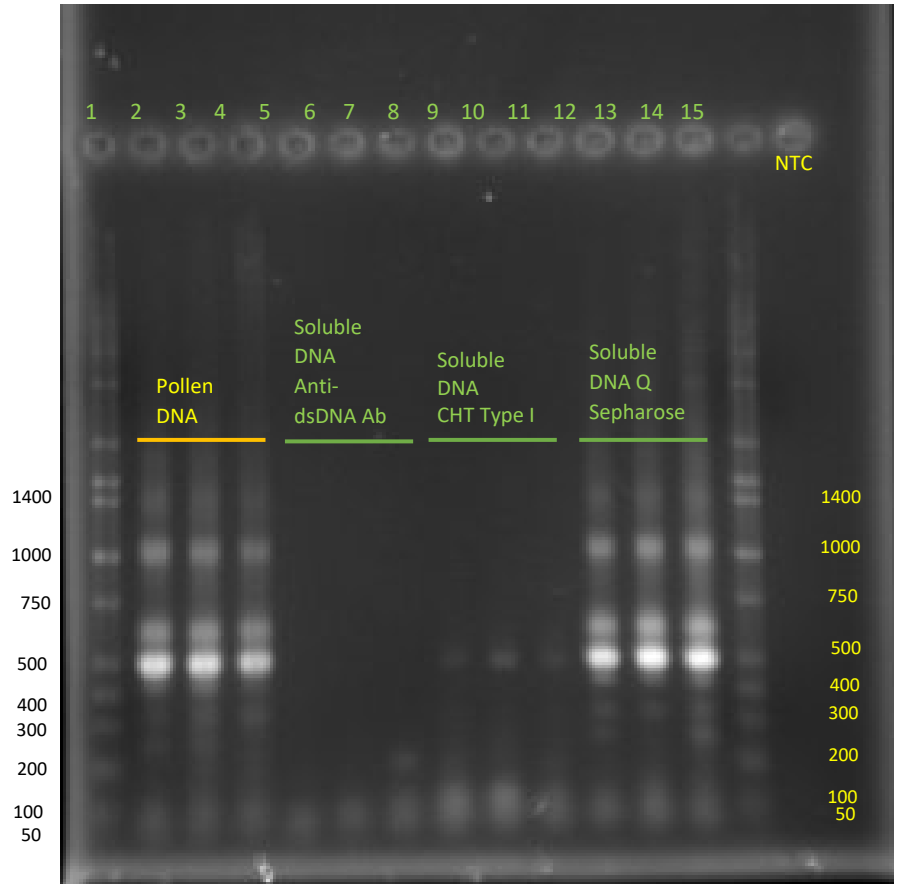


Figure 21. Agarose gel electrophoresis of ITS2.

Lane 1 and 14: DNA ladder
Lanes 2 to 4: ITS2 from pollen of raw honey
Lanes 5 to 13: ITS2 from pollen-free filtered honey captured using three methods-
a) anti-dsDNA Ab coupled to magnetic particles
b) CHT type I
c) Q Sepharose
Lane 15: No-template control

The ITS2 products obtained were then sequenced and plant species identified were analyzed by plotting a heatmap to understand the abundances of plant species seen. The unmerged forward reads obtained were compared to see if we were seeing the same plant species in ITS2 PCR products of pollen DNA and soluble DNA captured by the three methods, as shown in Figure 22. We saw similar plant species

between pollen DNA and soluble DNA content of the honey. A region-specific plant hit *Julgans major* was observed in all three replicates of pollen DNA and soluble DNA by Q Sepharose® and in one of the replicates by CHT.

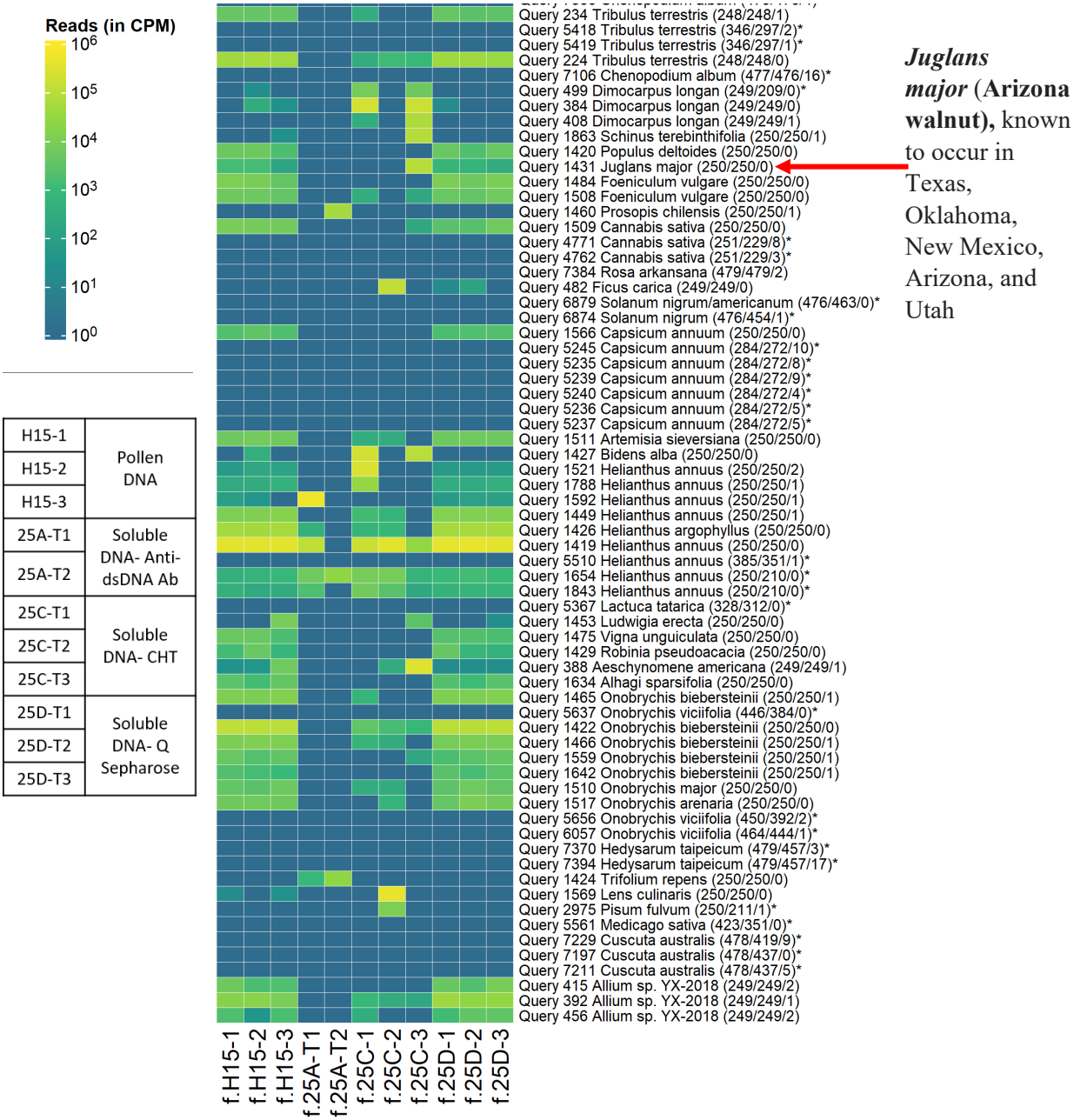


Figure 22. Heatmap for read count distribution and assigned taxa of ITS2 sequences obtained from pollen of raw honey and pollen-free filtered honey

Three numbers are printed after each taxonomic assignment. The first number refers to the ASV length, the second number indicates the portion of the ASV length that mapped to a GenBank sequence, and the third number is the number of mismatches. Taxonomic assignments where only a fraction of the ASV sequences matched to the NCBI nt database are marked with asterisks.

We also plotted rarefaction curves to study observed species richness in the three replicates of pollen DNA as shown in Figure 23. P1, P2, and P3 are three different pollen fractions that were processed to isolate plant gDNA and sequenced using NGS. The plot describes the number of different plants observed as a function of the total number of reads (sample size). If true diversity is so large that the analysis captures only a small fraction of species, rarefaction curves are straight with constant slope – i.e., doubling sequencing effort reveals twice as many species in an effectively inexhaustible pool. The flattening of the rarefaction curves we obtained shows that the majority of the diversity of the sample was captured in the sequencing of each aliquot taken for analysis.

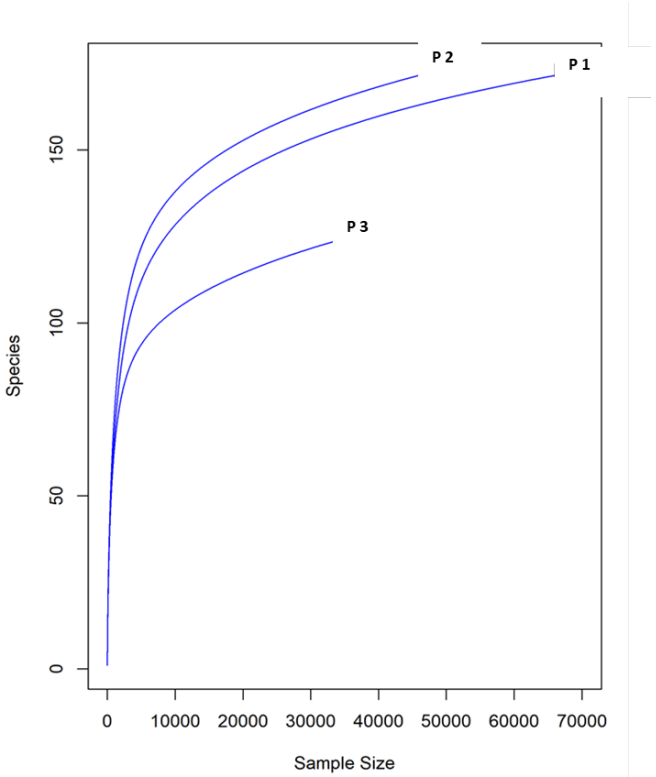


Figure 23. Rarefaction curves of ITS2 PCR products of triplicates of pollen DNA (unmerged forward reads).

We then explored the use of Q Sepharose to capture pollen-free plant gDNA from two additional honey samples H75 (Greek) and H58 (Argentina). We saw a similar pattern of ITS2 PCR products between pollen DNA and soluble DNA captured by Q Sepharose, as shown in Figure 24. The DNA sequencing results are in process.

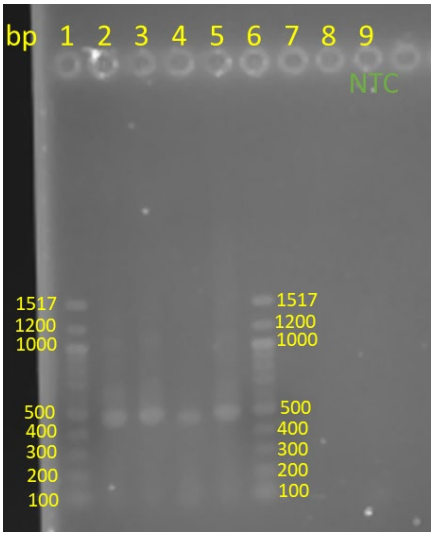


Figure 24. Agarose gel electrophoresis of ITS2 PCR products

Lane 1 and 6: DNA ladder
Lanes 2 and 3: Pollen DNA of H75 and H58 respectively
Lanes 4 and 5: Soluble DNA captured by Q Sepharose of H75 and H58 respectively
Lane 9: No-template control

4. CHALLENGING SAMPLES

The primary goal of this project was to develop a universal and robust method for the isolation, barcode PCR, sequencing and analysis of plant gDNA to facilitate the authentication of honey. The DNA isolation method used should provide good quality plant gDNA irrespective of honey, color, crystallization, age, and most importantly, pollen content. Maintaining the extracted plant gDNA intact also is crucial for efficient PCR amplification of ITS2. We used less-aggressive extraction methods to prevent the degradation of the DNA template throughout the isolation process. We incorporated the use of Proteinase K in the DNA extraction process (refer to section 2.3) to selectively degrade any contaminating proteins and nucleases present in pollen lysate. This step helped prevent the degradation of crude plant gDNA template.

We ran >300 samples with our first standard workflow, under pandemic restrictions and with (pandemic-induced) very slow turnround from our sequencing service provider. We obtained most of the data in this report in this fashion, but we also found that many samples gave poor results with this protocol (especially soluble DNA from filtered honey). The primary problem was in obtaining sufficient, good-quality plant gDNA template for PCR. Some samples gave either no specific product during PCR or poor reads after DNA sequencing. As we accumulated experience in handling the various sample types, we implemented an improved workflow to improve DNA template quality. We repeated the DNA extraction process as described in section 2.4 and introduced a new step of storing the extracted plant gDNA in Buffer AE (10 mM Tris-Cl, 0.5 mM EDTA; pH 9.0) overnight at -20^{o}C before further downstream analysis. We also implemented an additional silica column treatment to remove the PCR inhibitors present in the isolated gDNA template. For successful PCR, it was necessary to add sufficient DNA template, usually between 10-50 ng per PCR reaction. This work is ongoing using our own resources, and we now can achieve robust PCR amplification with a wide variety of samples, including many samples that previously gave only poorly-sequenceable amplicons. We expect to receive sequencing results from these improved methods over the next few weeks.

5. CONCLUSIONS

We report NGS sequencing and bioinformatic analysis of 303 honey samples, and the first methods for identifying the plant origins of ultra-filtered honey samples from soluble DNA. We have identified (a few) country- and region-specific plant DNA sequences. The COVID-19 pandemic imposed constraints on sample collection and provenance, and this exploratory project involved a relatively small number of samples, but the results are encouraging and suggest that with more work this technique could be relatively effective. We demonstrated efficient testing of claims of honey origin, e.g., for validating Manuka honey for which false claims appear to be quite common. A significant fraction of samples was not effectively processed to meet our high level of stringency with our initial protocols, and we did not have time and lab access enough under the pandemic to resolve this, but we believe the more-robust purification methods we have developed will allow effective analysis of the great majority of honey samples (more samples are currently out for sequencing, using our own resources). We believe the strategies developed for DNA capture and the bioinformatic pipeline can also be applied to identify the origins of difficult and degraded DNA templates for many other forensic applications, as illustrated in Figure 25.

DNA sequences obtained from this project will increase the richness of the public DNA database and help link occurrences of source plants across the world. Thus, by blending efforts in DNA purification and sequencing, we have established techniques that will help mitigate fraud associated with honey imports and indirectly aid in providing authentic and safe food for consumers.

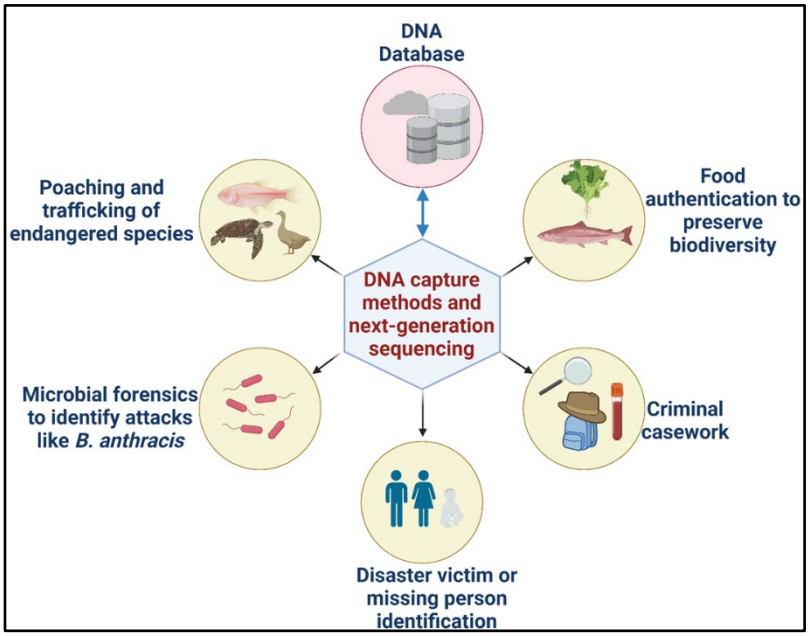


Figure 25. Broader impact of DNA capture methods for other applications

6. BIBLIOGRAPHY

1. McConnell, M. & Bond, J. K. *Sugar and Sweeteners Outlook*. *US Department of Agriculture, Economic Research Service* (2020).

2. Bryant, V. M., Jones, J. G., Mildenhall, D. C. & Jones, J. G. Forensic Palynology in the United States of America. **14**, 193–208 (1990).

3. FENTANYL: How Pollen Analysis Can Help. https://www.cbp.gov/sites/default/files/assets/documents/2019-Mar/fentanyl-factsheet.pdf.

4. Bell, K. L., Burgess, K. S., Okamoto, K. C., Aranda, R. & Brosi, B. J. Review and future prospects for DNA barcoding methods in forensic palynology. *Forensic Science International: Genetics* **21**, 110–116 (2016).

5. Bell, K. L. *et al.* Pollen DNA barcoding: current applications and future prospects. *Genome* **59**, 629–640 (2016).

6. Torricelli, M., Pierboni, E., Tovo, G. R., Curcio, L. & Rondini, C. In-house Validation of a DNA Extraction Protocol from Honey and Bee Pollen and Analysis in Fast Real-Time PCR of Commercial Honey Samples Using a Knowledge-Based Approach. *Food Anal. Methods* **9**, 3439–3450 (2016).

7. Jain, S. A., Jesus, F. T. de, Marchioro, G. M. & Araújo, E. D. Extraction of DNA from honey and its amplification by PCR for botanical identification. *Food Sci. Technol.* **33**, 753–756 (2013).

8. Lalhmangaihi, R., Ghatak, S., Laha, R., Gurusubramanian, G. & Kumar, N. S. Protocol for optimal quality and quantity pollen DNA isolation from honey samples. *J. Biomol. Tech.* **25**, 92–95 (2014).

9. Cheng, H. *et al.* Isolation and PCR Detection of Foreign DNA Sequences in Bee Honey Raised on Genetically Modified Bt (Cry 1 Ac) Cotton. *Food and Bioproducts Processing* **85**, 141–145 (2007).

10. CBP Trade Enforcement - Operational Approach | U.S. Customs and Border Protection. https://www.cbp.gov/document/fact-sheets/cbp-trade-enforcement-operational-approach LB - l12kV (2019).

11. Poczai, P. & Hyvönen, J. Nuclear ribosomal spacer regions in plant phylogenetics: problems and prospects. *Molecular Biology Reports* **37**, 1897–1912 (2010).

12. Han, J. *et al.* The Short ITS2 Sequence Serves as an Efficient Taxonomic Sequence Tag in Comparison with the Full-Length ITS. *BioMed Research International* **2013**, 1–7 (2013).

13. Chen, S. *et al.* Validation of the ITS2 Region as a Novel DNA Barcode for Identifying Medicinal Plant Species. *PLoS ONE* **5**, e8613 (2010).

14. Yao, H. *et al.* Use of ITS2 region as the universal DNA barcode for plants and animals. *PLoS ONE* **5**, (2010).

15. Soares, S., Amaral, J. S., Oliveira, M. B. P. P. & Mafra, I. Improving DNA isolation from honey for the botanical origin identification. *Food Control* **48**, 130–136 (2015).

16. Goux, H. J., Chavan, D., Crum, M., Kourentzi, K. & Willson, R. C. Akkermansia muciniphila as a Model Case for the Development of an Improved Quantitative RPA Microbiome Assay. *Front. Cell. Infect. Microbiol.* **8**, 237 (2018).

17. Callahan, B. J. DADA2 ITS Pipeline Workflow (1.8). https://benjjneb.github.io/dada2/ITS_workflow.html (2021).

18. Callahan, B. J., Sankaran, K., Fukuyama, J. A., McMurdie, P. J. & Holmes, S. P. Bioconductor workflow for microbiome data analysis: from raw reads to community analyses [version 2; peer review: 3 approved]. *F1000Research* **5**, 1492 (2016).

19. Gaspar, J. M. NGmerge: Merging paired-end reads via novel empirically-derived models of sequencing errors. *BMC Bioinformatics* **19**, 1–9 (2018).

20. Callahan, B. J. *et al.* DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods* **13**, 581–583 (2016).

21. Rosemary, J. M., Dunn, J. C. & Martine, G. New universal ITS2 primers for high-resolution herbivory analyses using DNA metabarcoding in both tropical and temperate zones. *Scientific Reports* **8**, 1–15 (2018).

22. Timpano, E. K., Scheible, M. K. R. & Meiklejohn, K. A. Optimization of the second internal transcribed spacer (ITS2) for characterizing land plants from soil. *PLOS ONE* **15**, e0231436 (2020).

Table 2. Details of Manuka Samples

Sample No	NGS Sample Code	Name on bottle	Honey plant name	Type of honey	Manufactured by	Packed by	Product of	Qty (g)	Cost ($)
17	H19	Wedderspoon Gold	Manuka Honey	Raw and unpasteurized honey	-	Distributed by-Wedderspoon Organics, USA	New Zealand	325	22.95
163	H42	Comvita	Manuka Honey	Raw, Wild & unpasteurized	Comvita New Zealand Ltd, Bay of plenty, New Zealand	-	New Zealand	500	36.99
223	H150	Wedderspoon	Monofloral Manuka honey	Raw	Wedderspoon organic New Zealand 621 Linesid road, Rangiora, NZ 7400	Distributed by-Wedderspoon organic USA, LLC, Malvern, PA	New Zealand	1000	76.99
224	H151	By Buzzz	Manuka (Lfactor 18)	Coarse filtered, unpasteurized, non-GMO	-	Distributed by Buzzz PTY LTD, Australia	Australia	500	35.47
225	H111	Manuka Health	Manuka	Raw & Unpasteurized	Manuka Health New Zealand Lts, Te Awamutu, New Zealand	Manuka Health New Zealand Lts, Te Awamutu, New Zealand	New Zealand	500	100.00
226	H108	Manuka Doctor	Manuka Multifloral	-	-	Distributed by Liberty Richard, a division of World Finer Foods	New Zealand	250	20.08
227	H112	Manuka Doctor	Manuka Monofloral	-	-	Distributed by Liberty Richard, a division of World Finer Foods, Bloomfield, NJ, USA	New Zealand	250	39.03

Sample No	NGS Sample Code	Name on bottle	Honey plant name	Type of honey	Manufactured by	Packed by	Product of	Qty (g)	Cost ($)
228	H113	Manuka Doctor	Manuka Monofloral	-	-	Distributed by Liberty Richard, a division of World Finer Foods, Bloomfield, NJ, USA	New Zealand	250	32.29
229	H114	Wilderness Valley	Manuka	-	Wilderness Valley LTD, Auckland, NZ	-	New Zealand	250	21.99
230	H115	Manuka Health	Manuka (Monofloral)	Raw & Unpasteurized	Manuka Health NZ, Te Awamutu, NZ	-	New Zealand	250	16.49
231	H116	Taylor Pass Honey Co.	multifloral manuka	-	Taylor Pass Honey Company, Blenheim, NZ	-	New Zealand	250	35.75
232	H117	Nature's Gold	manuka honey	100% raw pure honey	Honeybiz Australia Pty Ltd, Taringa QLD, Australia (distributed by and manufactured by)	-	Australia	250	23.95
233	H118	Bee's Inn	Manuka	100% Pure New Zealand Honey	-	Bee's Inn Manuka Ltd, Ohaupo, NZ	New Zealand	250	39.99
234	H119	WildCape	Manuka	-	-	Savage Horticulture Ltd, New Zealand	New Zealand	250	37.89
235	H120	Puhoi Honey	Manuka (multifloral)	-	Puhoi Honey Ltd, New Zealand	-	New Zealand	500	33.99

Sample No	NGS Sample Code	Name on bottle	Honey plant name	Type of honey	Manufactured by	Packed by	Product of	Qty (g)	Cost ($)
236	H104	Bees Knees Honey Co.	Manuka	-	-	-	-	-	24.99
237	H122	Good Natured	Manuka	cold extraction, raw	Australia's Manuka, Tyagarah, NSW, Australia	Distributed by Good Natured pty Ltd, San Rafael CA	Australia	250	52.88
238	H79	Pacific Resources International (PRI)	Manuka	Raw	-	PRI, CA, USA	New Zealand	500	15.00
239	H80	Forest Gold	Manuka	-	Forest Gold Ltd., New Zealand	-	New Zealand	250	45.00
240	H123	Forest Gold	Manuka	-	Forest Gold Ltd., New Zealand	-	New Zealand	250	85.00
241	H81	Manukora	Manuka (multifloral)	Raw, non-GMO	-	Ora Group Ltd, (one bottle - NZ, another bottle - LA, California)	New Zealand	250	14.99
242	H82	Kiva	Manuka	Raw	-	Kiva Health 2002 Honolulu, HI	New Zealand	250	64.99
243	H124	ManukaGuard	Manuka	Raw	-	bottled in the USA using wild harvested authentic manuka honey from New Zealand	USA/ New Zealand	250	15.04
244	H83	Egmont Honey	Manuka (from around Mt Egmont)	Raw	Egmont Honey LTD, NZ	-	New Zealand	500	37.99

Sample No	NGS Sample Code	Name on bottle	Honey plant name	Type of honey	Manufactured by	Packed by	Product of	Qty (g)	Cost ($)
245	H84	Mossop's	Manuka	-	-	Mossop's. SH 29, Tauranga, NZ	New Zealand	250	17.70
246	H85	Ruapehua	Manuka (multifloral)	unfiltered	Custard Square Health Limited, New Zealand	-	New Zealand	110	12.75
247	H86	Wedderspoon	Manuka (multifloral)	Non-GMO, raw, unpasteurized, antibiotics-free	Wedderspoon Organic New Zealand, Rangiora, NZ	Distributed by Wedderspoon Organic INC, Duncan BC, Canada	New Zealand	250	26.99
248	H109	Taku Honey	Manuka blend	-	-	-	New Zealand		15.99
249	H110	Happy Valley	Manuka	-	Happy Valley Honey (NZ) Ltd, Papakura NZ	-	New Zealand	500	33.99
250	H87	1839 Honey	Manuka	Triple churned	1839 Ltd. Tariko, Tauranga, NZ	-	New Zealand	500	49.99
251	H88	Steens	manuka	Raw, cold pressed, GMO free	PA & SC Steens Limited, NZ	Steens Honey Ltd, UK	New Zealand	500	146.99
252	H89	New Zealand Honey Co.	Manuka (Monofloral)	Raw, Non-GMO, Pure	New Zealand Honey Co., Ltd, Wanaka NZ	-	New Zealand	250	24.87
253	H90	Three Peaks	Manuka	-	Pure Manuka Honey Ltd, NZ	-	New Zealand	250	24.99
254	H125	Zealandia	Manuka Monofloral	-	Zealandia Honey Limited, Waikato, NZ	-	New Zealand	250	44.99
255	H126	Nelson honey (bee zesty)	manuka	GMO free	Nelson Honey & Marketing (NZ) Ltd, NZ	Distributed by Nelson Honey UK Ltd, UK	New Zealand	250	25.97

Sample No	NGS Sample Code	Name on bottle	Honey plant name	Type of honey	Manufactured by	Packed by	Product of	Qty (g)	Cost ($)
256	H127	Avatar	Manuka (Wairarapa coast)	-	Avatar Honey NZ Ltd, Wairarapa, NZ	-	New Zealand	250	110.00
286	H105	Airborne	Manuka blend	-	-	packed by Airborne Honey Limited, NZ	New Zealand	500	29.95
292	H106	Tahi	Manuka (UMF 5+)	Carbon neutral honey, GE Free, Natural	Tahi, Pataua, RD, NZ	-	New Zealand	400	39.00
299	H107	Tahi	Manuka (multifloral) biodiversity positive honey	Carbon neutral honey, GE Free, Natural	Tahi, Pataua, RD, NZ	-	New Zealand	400	23.00

Characterization of ITS2 sequences from honey

April 25, 2021

```
# Setup
if (!require("pacman")) install.packages("pacman")
pacman::p_install_gh("rlbarter/superheat")
pacman::p_load("knitr","dada2","kableExtra","ShortRead","doFuture","pbapply","superheat","future.apply",
               "Rtsne", "doRNG","readxl","zip","RColorBrewer","wordcloud","mgsub","Biostrings",
               "htmlwidgets","htmltools")
opts_knit$set(root.dir = "~/R/Dimple/DNAbarcoding/honeyITS2/doc/")

def.chunk.hook  <- knitr::knit_hooks$get("chunk")
knitr::knit_hooks$set(chunk = function(x, options) {
  x <- def.chunk.hook(x, options)
  ifelse(options$size != "normalsize", paste0("\n \\", options$size,"\n\n", x, "\n\n \\normalsize"), x)
})

opts_chunk$set(cache = TRUE, echo = TRUE, fig.align = "center", fig.width = 12, cache.lazy = TRUE,
               fig.height = 6, message = FALSE, warning = FALSE, eval = FALSE, dpi = 150, size = "footnotesize")

setwd(opts_knit$get("root.dir"))

source("../src/my_save.R")

write_bib(names(sessionInfo()$otherPkgs), file = "references.bib")

options(citation_format = "pandoc", tikzLatexPackages = c(getOption("tikzLatexPackages")),
        Biostrings.coloring=FALSE)
```

Data

Datasets are DNA sequences obtained from honey samples collected from various parts of the world.

```
# Load and reformat sample metadata
mdat <- list(readxl::read_excel("../data/Honey sample country and continent.xlsx", range = "A5:E500"),
             readxl::read_excel("../data/Honey sample country and continent.xlsx", range = "G5:K500"))
mdat <- do.call(rbind, lapply(mdat, as.data.frame))
mdat <- mdat[rowSums(is.na(mdat)) == 0,]
mdat[mdat$Code == "H9","Description"] <- "Sep-18"
ord <- order(mdat$AltCode)
mdat <- mdat[ord,]
rownames(mdat) <- NULL

dir.create("../Supplementary Materials", showWarnings = F)
write.csv(mdat, "../Supplementary Materials/S1.csv", row.names = F)
```

Sample metadata available in **S1.csv**.

Workflow

An initial filtering was executed to remove paired-end reads flanked by mismatched primers. The bioinformatic analysis was then adapted from the DADA2 ITS Pipeline Workflow and the workflow for Microbiome Data Analysis.

The quality of the raw sequences were examined.

```
# Get file paths to sequences
fls <- list.files("../data", recursive = T, pattern = "fastq.gz", full.names = T)
ord <- mgsub::mgsub(basename(fls), c("^.*(H\\d+_R[12]).+", "(\\d+[A-Z]*_R[12]).+"), c("\\1", "\\1"))
ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
ord <- order(ord)
fls <- fls[ord]
ids <- unique(gsub("_R[12].*", "", basename(fls)))
ids <- gsub("DC0[123]-", "", ids)

# Fix plotQualityProfile's behavior of arranging plots alphabetically
body(plotQualityProfile)[[6]] <- quote({anndf$minScore <- min(anndf$minScore)
                                        plotdf$file <- factor(plotdf$file, levels = unique(plotdf$file))
                                        statdf$file <- factor(statdf$file, levels = unique(statdf$file))
                                        anndf$file <- factor(anndf$file, levels = unique(anndf$file))
                                        })

# Examine quality scores
loc <- "../results/fig/plotQualityProfileInitial/"
dir.create(loc, showWarnings = F)
for (i in ids) {
  fl.loc <- paste0(loc, "plotQualityProfileInitial-", i, ".png")
  if (!file.exists(fl.loc)) {
    png(fl.loc, height = 3, width = 8, res = 300, units = "in")
    print(plotQualityProfile(grep(paste0(i, "_"), fls, value = T), n = 2E6) + facet_wrap(~file, ncol = 2))
    dev.off()
  }
}

zip("../Supplementary Materials/S2.zip",
    files = list.files("../results/fig/plotQualityProfileInitial/", full.names = T),
    include_directories = F, mode = "cherry-pick")
```

Quality profiles for all samples are available in **S2.zip**.

Primer-mismatch filtering

A cursory inspection revealed that each FASTQ file is a mixture of both forward and reverse reads. Furthermore, some reads do not seem to match the ITS2 primer sequences used. Below are shown the six FASTQ files containing PE reads corresponding to the three honey samples as well as the first three reads from FASTQ file DC01-H1-R1_001. Highlighted in yellow are the bases corresponding to the primer sequences. The 20-nt forward primer sequence can be found in the first read which is to be expected for a FASTQ file of forward reads (R1_001), but the second read shows that it is preceded by the 21-nt reverse primer sequence. The third sequence is an example of a read that is preceded by neither of the two primer sequences. Paired-end reads that are preceded by either primer exceeding a mismatch threshold are taken to be unreliable and thus removed.

```
root@00_fastq# ls
DC01-H1_R1_001.fastq.gz  DC02-H2_R1_001.fastq.gz  DC03-H3_R1_001.fastq.gz  md5sum_list.txt
DC01-H1_R2_001.fastq.gz  DC02-H2_R2_001.fastq.gz  DC03-H3_R2_001.fastq.gz
root@00_fastq# zcat DC01-H1_R1_001.fastq.gz | head -n 10
@M02713:182:GW190901AmpliconEZCycle0821:1:2105:16454:1390 1:N:0:GCAGAAGA+ACCTGGTT
ATGCGATACTTGGTGTGAATTGCAGAATCCCGTGAACCATCGAGTTTTTGAACGCAAGTTGCGCCCGAAGCCATTCGGTTGAGGGCACGTCTGCCTGGGCGTCACGCATCGCTGCCCCCCCCCCCCCACAAAACACC
CCATGCGGGCTCGTTGCGTAGGCAGGGGCACCCGCTGGCCTCCCGTGCGCACCGTCGGGCGGATGGCTTAAATTCGATTCCTCGGCGCACGTCGTCGCGACACTACGGTGGTTG
+
BAAAB3>ABFFFGBFGEFGFGGHHHHHHHHGGDGGGHHHHHFBFGFHHGGGHHHGGGGHFHHGGGGEGGGGHHHHHHHGGGFFHGG1EGFEFGHEHHFGGGGGGGDHGGGDGHGGHHGGGCGGGGG-...;C?EGG
GGGGFFDC-->.;9..9>;.@A//:BF-9-9..9>-;-AF/AEAD...-9-9--9.--;-9---99.9.BF/FF/.///;//;A..-9-9@..--.-9----99//;A.;9:.A
@M02713:182:GW190901AmpliconEZCycle0821:1:2105:17434:1552 1:N:0:GCAGAAGA+ACCTGGTT
GACGCTTCTCCAGACTACAATTCGGACGACGAAGCCGCCCGATTTTCAAGCTGGGCTGTTCCCGGTTCGCTCGCCGTTACTAAGGGAATCCTAGTAAGTTTCTTTTCCTCCGCTTATTGATATGCTTAAACTCAGC
GGGTAGTCCCGCCTGACCTGGGGTCGCGATCGAAGCATCGTCATAAGACAACACATCAGGGTACTTTAAGAGCATCTACCCTCAAGAAAGTAAACGCACGACACGAGACGACTG
+
AAABB?BBFFFFGGGGGGFGGGFGHGGGGGGGFFGGGGGGGGGGGHGHHHHHHHGGHGHHHHHGGGGGGGGHGGGE>EEGHFHHHGHGEHHHFHGHGGHHHHHHHHHHHHHGGGGGHHHHHHHGHHHGHHGHHHFH
GGGGGGHHGHGGGGHHHHHHHGGGGGGGGGGGGGGGGFFFFFFFFFFFFFFFFFFFFFF9FFFFFFFFFFFFFFFFFFFFFFFEFBFBFFFFFFFFFCFAFFFFAFFF
@M02713:182:GW190901AmpliconEZCycle0821:1:2105:13145:1578 1:N:0:GCAGAAGA+ACCTGGTT
CGATCGAAGCATCGTCATAAGACAACACATCAGGGTACTTTAAGAGCATCTACTCTCAAGAAAGTAAACGCACGACACGAGACGACTGTCTTATCAACCACCACTAGCCGTGCGTCCCTCGCGAGGAGACTCCTAT
TTAGGCCAACCACGCCATGGGCACGGGAGACCAATCTCCGCCCCAACACAAGACAGCCCTATAGGGGATGCCTGGTGGGGGCGACGTGATGCGTGACGCCCAGGCCGACGTGCC
root@00_fastq#
```

Primer sequences match in at most 8 bases Due to the nature of the sequencing platform used, all FASTQ files are random mixtures of forward and reverse reads instead of the typical case where R1_001 and R2_001 files containing forward and reverse reads respectively (from correspondence with GENEWIZ). The first 20 or 21 bases can be used to identify whether a read is forward or reverse based on its sequence matching with the primer sequences. However, sequencing errors can introduce mismatches to otherwise valid reads. On the other hand, relaxing the stringency in sequence matching too much will lead to misidentification of the reads.

The example below shows how the forward and reverse primer sequences align to satisfy the maximum number of mismatches specified.

```
> fwdPrimer
[1] "ATGCGATACTTGGTGTGAAT"
> matchPattern(fwdPrimer, revPrimer, max.mismatch = 12)
  Views on a 21-letter BString subject
subject: GACGCTTCTCCAGACTACAAT
views:
    start end width
[1]    -3  16    20 [    GACGCTTCTCCAGACT]
[2]     2  21    20 [ACGCTTCTCCAGACTACAAT]
> revPrimer
[1] "GACGCTTCTCCAGACTACAAT"
> matchPattern(revPrimer, fwdPrimer, max.mismatch = 13)
  Views on a 20-letter BString subject
subject: ATGCGATACTTGGTGTGAAT
views:
    start end width
[1]     0  20    21 [ ATGCGATACTTGGTGTGAAT]
[2]     5  25    21 [GATACTTGGTGTGAAT     ]
```

This procedure is repeated for various mismatch stringencies and the results can be found in **S3.html**.

```
fls <- list.files("../data", recursive = T, pattern = "fastq.gz", full.names = T)
ord <- mgsub::mgsub(basename(fls), c("^.*(H\\d+_R[12]).+", "(\\d+[A-Z]*_R[12]).+"), c("\\1", "\\1"))
ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
ord <- order(ord)
fls <- fls[ord]
dat <- pblapply(fls, readFastq)
names(dat) <- gsub(".fastq.gz", "", basename(fls))
names(dat) <- gsub("DC\\d+-", "", names(dat))
names(dat) <- gsub("_001", "", names(dat))
srs <- pblapply(dat, sread)


# Count primer mismatch frequency with respect to maximum number of mismatches
fwdPrimer <- "ATGCGATACTTGGTGTGAAT"
revPrimer <- "GACGCTTCTCCAGACTACAAT"

primerMatch <- function(primer, sreads) {
  # Determine frequency of unique reads
  seqs <- pbsapply(sreads, function(x) as.character(x[1:nchar(primer)]))
  seqs <- sort(table(seqs), decreasing = T)

  # Flag unique read if not exceeding maximum mismatch threshold
  seqs.match <- sapply(0:nchar(primer),
                       function(i) vmatchPattern(primer,
                                                 DNAStringSet(names(seqs)), max.mismatch = i))
  names(seqs.match) <- 0:nchar(primer)

  # Count total number of times primer map to unique reads passing threshold
  seqs.uniq <- sapply(as.character(0:nchar(primer)),
                      function(i) sum(lengths(seqs.match[[i]])))

  # Determine the total number of reads passing threshold
  seqs.tot <- sapply(as.character(0:nchar(primer)),
                     function(i) {
                       idx <- as.logical(lengths(seqs.match[[i]]))
                       return(c(sum(idx), sum(seqs[idx])))
                     })
  seqs.tot <- t(seqs.tot)
```

```
  return(data.frame(`total number of ways primer mapped to unique reads` = seqs.uniq,
                    `total number of unique reads matched` = seqs.tot[,1],
                    `total number of reads matched` = seqs.tot[,2]))
}

# Determine at which maximum mismatch threshold value the forward primer sequence starts match with
# the reverse primer and vice-versa
tbl.mismatch <- merge(primerMatch(fwdPrimer, revPrimer),
                      primerMatch(revPrimer, fwdPrimer), by = 0, all = T)
tbl.mismatch <- tbl.mismatch[order(as.numeric(tbl.mismatch$Row.names)),]
row.names(tbl.mismatch) <- NULL

my_save(tbl.mismatch, file = "../results/tbl.mismatch.Rds", overwrite = T)

# Count mismatches across FASTQ files
loc <- "../results/srsMatch/"
dir.create(loc, showWarnings = F)

idx <- tools::file_path_sans_ext(list.files('../results/srsMatch/'))
srs <- srs[!names(srs) %in% idx]

if (length(srs) != 0) {
  registerDoFuture()
  options(future.globals.maxSize = 20*1024^3)
  plan(multiprocess, workers = availableCores() - 1)

  foreach(i = 1:length(srs)) %dopar% {
    x <- primerMatch(srs[[i]], primer = ifelse(grepl("_R1$", names(srs[i])), fwdPrimer, revPrimer))
    saveRDS(x, file = paste0(loc, names(srs[i]), ".Rds"))
  }

  plan(sequential)
}
# Tabulate primer mismatch stats
tbl <- readRDS("../results/tbl.mismatch.Rds")
tbl <- tbl[,c(1, 2, 5, 3, 6, 4, 7)]

kbl(format(tbl, big.mark = ","), format = 'html', align = "c", booktabs = T,
      col.names = c("No. of \n mismatches", rep(c("forward primer", "reverse primer"), 3)),
      caption = paste0("Mapping of the primer sequences to each other as a function of the maximum number",
                       "of mismatches allowed", collapse = "") %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  column_spec(1, bold = T) %>%
  add_header_above(c(" " = 1, "Total number of ways primer\nmapped to unique reads" = 2,
                     "Total number of unique\nreads matched" = 2,
                     "Total number of reads matched" = 2)) %>%
  cat(., file = "../Supplementary Materials/S3.html")
```

Primer mismatch errors in NGS reads are kept at 5 nt as a compromise between read quality and abundance The same procedure as with the primer sequences is applied to the FASTQ files on full-length reads and the results are tabulated for each primer sequence.

- Mapping of the forward primer sequence to the first 20 bases of the FASTQ reads as a function of the maximum number of mismatches allowed (**S4.html**)

```
loc <- "../results/srsMatch/"
fls <- list.files(loc, full.names = T)
ord <- tools::file_path_sans_ext(basename(fls))
ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
ord <- order(ord)
fls <- fls[ord]

sm <- pblapply(fls, readRDS)
names(sm) <- tools::file_path_sans_ext(basename(fls))
```

```
tbl <- do.call(cbind, lapply(1:3, function(i) do.call(cbind, lapply(sm[grepl("_R1", names(sm))], "[[", i))))
tbl <- data.frame(`No. of mismatches` = 1:nrow(tbl) - 1, tbl)
rownames(tbl) <- NULL
names(tbl) <- gsub("X", "", names(tbl))

sketch  <-  htmltools::withTags(table(
  class = 'display',
  thead(
    tr(
      th(rowspan = 2, style = "border-right: solid 2px;", 'No. of mismatches'),
      th(colspan = (ncol(tbl) - 1)/3,
         style = "border-right: solid 2px;", 'Total number of ways primer mapped to unique reads'),
      th(colspan = (ncol(tbl) - 1)/3,
         style = "border-right: solid 2px;", 'Total number of unique reads matched'),
      th(colspan = (ncol(tbl) - 1)/3,
         style = "border-right: solid 2px;", 'Total number of reads matched')
    ),
    tr(
      lapply(gsub("_(?=R)", "\n", gsub("\\.\\d$", "", colnames(tbl))[-1], perl = T),
             function(x) {
               if (grepl(paste0(gsub("_.*", "", tail(names(tbl), 1)), "\nR1"), x))
                 th(style = "border-right: solid 2px;", x)
               else
                 th(x)
             })
    )
  )
))

datatable(tbl, container = sketch, rownames = F,
          extensions = c('FixedColumns','Buttons'),
          options = list(columnDefs = list(list(className = 'dt-right', targets = "_all"),
                                           list(className = 'dt-head-center', targets = "_all")),
                         scrollX = TRUE, scrollY = FALSE, paging = TRUE,
                         dom = 'Blfrtip', lengthMenu = seq(10, 10*ceiling(nrow(tbl)/10), 10),
                         buttons = c('copy','excel','print'),
                         fixedColumns = list(leftColumns = 1, rightColumns = 0))) %>%
  formatStyle(colnames(tbl), `font-size` = '13px') %>%
  formatStyle(1, `white-space` = 'nowrap', fontWeight = 'bold', `text-align` = 'center') %>%
  formatStyle(1 + ((ncol(tbl) - 1)/3 * 0:3), `border-right` = "solid 2px") %>%
  formatCurrency(colnames(tbl), currency = "", interval = 3, mark = ",", digits = 0) %>%
  htmlwidgets::saveWidget(., "../Supplementary Materials/S4.html")
```

- Mapping of the reverse primer sequence to the first 21 bases of the FASTQ reads as a function of the maximum number of mismatches allowed (**S5.html**)

```
tbl <- do.call(cbind, lapply(1:3, function(i) do.call(cbind, lapply(sm[grepl("_R2", names(sm))], "[[", i))))
tbl <- data.frame(`No. of mismatches` = 1:nrow(tbl) - 1, tbl)
rownames(tbl) <- NULL
names(tbl) <- gsub("X", "", names(tbl))

sketch  <-  htmltools::withTags(table(
  class = 'display',
  thead(
    tr(
      th(rowspan = 2, style = "border-right: solid 2px;", 'No. of mismatches'),
      th(colspan = (ncol(tbl) - 1)/3,
         style = "border-right: solid 2px;", 'Total number of ways primer mapped to unique reads'),
      th(colspan = (ncol(tbl) - 1)/3,
         style = "border-right: solid 2px;", 'Total number of unique reads matched'),
      th(colspan = (ncol(tbl) - 1)/3,
         style = "border-right: solid 2px;", 'Total number of reads matched')
    ),
    tr(
      lapply(gsub("_(?=R)", "\n", gsub("\\.\\d$", "", colnames(tbl))[-1], perl = T),
             function(x) {
               if (grepl(paste0(gsub("_.*", "", tail(names(tbl), 1)), "\nR2"), x))
```

```
                th(style = "border-right: solid 2px;", x)
              else
                th(x)
            })
    )
  )
))

datatable(tbl, container = sketch, rownames = F,
          extensions = c('FixedColumns','Buttons'),
          options = list(columnDefs = list(list(className = 'dt-right', targets = "_all"),
                                           list(className = 'dt-head-center', targets = "_all")),
                         scrollX = TRUE, scrollY = FALSE, paging = TRUE,
                         dom = 'Blfrtip', lengthMenu = seq(10, 10*ceiling(nrow(tbl)/10), 10),
                         buttons = c('copy','excel','print'),
                         fixedColumns = list(leftColumns = 1, rightColumns = 0))) %>%
  formatStyle(colnames(tbl), `font-size` = '13px') %>%
  formatStyle(1, `white-space` = 'nowrap', fontWeight = 'bold', `text-align` = 'center') %>%
  formatStyle(1 + ((ncol(tbl) - 1)/3 * 0:3), `border-right` = "solid 2px") %>%
  formatCurrency(colnames(tbl), currency = "", interval = 3, mark = ",", digits = 0) %>%
  htmlwidgets::saveWidget(., "../Supplementary Materials/S5.html")
```

Since misidentification starts to occur at a maximum mismatch of 12 or 13 between the two primer sequences, cutoffs can be chosen just below these two values for the FASTQ reads (*S3.html*). From *S4.html* and *S5.html*, this assertion is supported by the tabulated FASTQ read mappings based on the *total number of reads matched*. However, it is argued that FASTQ reads that match at high maximum mismatch thresholds would correspondingly have high error rates. Also, from the column of *total number of reads matched*, the increases are incremental for the *number of mismatches* ranging from 1 to 11 or 12. As such, based on the column of *total number of unique reads matched*, a cutoff of 5 for the maximum number of mismatches was selected.

```
# Filter out PE reads that have more than 5 mismatches on either primer sequence preceding it
primerCheck <- function(srs, fwdPrimer, revPrimer, numMismatch = 0) {
  fwdPrimer <- as.matrix(DNAString(fwdPrimer))
  revPrimer <- as.matrix(DNAString(revPrimer))

  primerMap <- sapply(srs, function(x) {
    if (sum(as.matrix(x)[1:length(fwdPrimer),] != fwdPrimer) <= numMismatch) {
      statusF <- 1
    } else {
      statusF <- 0
    }

    if (sum(as.matrix(x)[1:length(revPrimer),] != revPrimer) <= numMismatch) {
      statusR <- -1
    } else {
      statusR <- 0
    }

    if (statusF != statusR) {
      return(statusF + statusR)
    } else {
      return(NA)
    }

  })

  return(primerMap)

}

# Count mismatches across FASTQ files
loc <- "../results/pmMatch/"
dir.create(loc, showWarnings = F)

if (!exists("dat")) {
  fls <- list.files("../data", recursive = T, pattern = "fastq.gz", full.names = T)
```

```
  ord <- mgsub::mgsub(basename(fls), c("^.*(H\\d+_R[12]).+", "(\\d+[A-Z]*_R[12]).+"), c("\\1", "\\1"))
  ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
  ord <- order(ord)
  fls <- fls[ord]
  dat <- pblapply(fls, readFastq)
  names(dat) <- gsub(".fastq.gz", "", basename(fls))
  names(dat) <- gsub("DC\\d+-", "", names(dat))
  names(dat) <- gsub("_001", "", names(dat))
}

idx <- tools::file_path_sans_ext(list.files('../results/pmMatch/'))
srs <- pblapply(dat[!names(dat) %in% idx], sread)


if (length(srs) != 0) {
  registerDoFuture()
  options(future.globals.maxSize = 20*1024^3)
  plan(multiprocess, workers = availableCores() - 1)

  foreach(i = 1:length(srs)) %dopar% {
    x <- primerCheck(srs[[i]], fwdPrimer = fwdPrimer, revPrimer = revPrimer, numMismatch = 5)
    saveRDS(x, file = paste0(loc, names(srs[i]), ".Rds"))
  }

  plan(sequential)
}

fls <- list.files(loc, full.names = T)
ord <- tools::file_path_sans_ext(basename(fls))
ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
ord <- order(ord)
fls <- fls[ord]
pm <- pblapply(fls, readRDS)
names(pm) <- tools::file_path_sans_ext(basename(fls))

idx <- mapply(function(x, y) x + y,
              pm[grep("R1$", names(pm))], pm[grep("R2$", names(pm))], SIMPLIFY = F)
names(idx) <- gsub("_R.+", "", names(idx))

frs <- lapply(names(dat), function(i) dat[[i]][which(idx[[gsub("_.*$", "", i)]] == 0)])
names(frs) <- names(dat)
fm <- lapply(names(pm), function(i) pm[[i]][which(idx[[gsub("_.*$", "", i)]] == 0)])
names(fm) <- names(pm)

# Segregate forward and reverse reads into corresponding R1_001 and R2_001 containers
primerSwap <- function(frs1, frs2, fm) {
  fwd <- frs1
  rev <- frs2

  idx <- unique(gsub("_R.*", "", names(fm)))

  for (i in idx) {
    f.idx <- paste0(i, "_R1")
    r.idx <- paste0(i, "_R2")

    rev[[r.idx]][fm[[r.idx]] == 1] <- frs1[[f.idx]][fm[[f.idx]] == -1]
    fwd[[f.idx]][fm[[f.idx]] == -1] <- frs2[[r.idx]][fm[[r.idx]] == 1]
  }

  return(list(FWD = fwd, REV = rev))
}

swp <- primerSwap(frs[grep("R1$", names(frs))], frs[grep("R2$", names(frs))], fm = fm)
swp <- unlist(swp, recursive = F)
names(swp) <- gsub("^.+\\.", "", names(swp))

loc <- "../results/filtFASTQ/"
```

```
dir.create(loc, showWarnings = F)

for (i in names(swp)) {
  fl <- paste0(loc, i, ".fastq.gz")
  if (!file.exists(fl)) {
    writeFastq(swp[[i]], file = fl)
  }
}

# Save a sample of reads to file
swpReads <- R.utils::captureOutput(lapply(swp, sread))
writeLines(print(swpReads, quote = F), file("../Supplementary Materials/S6.txt"))
closeAllConnections()

# Get file paths to sequences
fls <- list.files("../results/filtFASTQ/", recursive = F, pattern = "fastq.gz", full.names = T)
ids <- gsub("_R.+", "", basename(fls))
ids <- unique(ids)

# Examine quality scores
loc <- "../results/fig/plotQualityProfileSwapped/"
dir.create(loc, showWarnings = F)
for (i in ids) {
  fl.loc <- paste0(loc, "plotQualityProfileswapped-", i, ".png")
  fl.src <- grep(paste0(i, "_"), fls, value = T)
  if (!file.exists(fl.loc) & all(file.info(fl.src)["size"] != 20)) {
    png(fl.loc, height = 3, width = 8, res = 300, units = "in")
    print(plotQualityProfile(fl.src, n = 2E6) + facet_wrap(~file, ncol = 2))
    dev.off()
  }
}

zip("../Supplementary Materials/S7.zip",
    files = list.files("../results/fig/plotQualityProfileSwapped/", full.names = T),
    include_directories = F, mode = "cherry-pick")
```

PE reads that have more than 5 mismatches are filtered out and forward and reverse reads were segregated to their corresponding *R1_001* and *R2_001* files respectively as the forward and reverse reads are mixed together in the original FASTQ files. Viewing a few reads from each segregated sample shows similar if not the same primer sequences at the start of each read (**S6.txt**). The quality profiles of the segregated reads can be found in **S7.zip**.

ITS DADA2 pipeline

NGS samples contain paired-end sequences that do not overlap The table below confirms that only the primers in their forward and reverse complement positions can be detected (allowing for a maximum mismatch of 5). Some ITS2 sequences exceed the maximum effective read length which is 480 nt (2×250 read length – 20 nt minimum overlap).

```
fwdPrimer <- "ATGCGATACTTGGTGTGAAT"
revPrimer <- "GACGCTTCTCCAGACTACAAT"

# Tabulate all possible primer orientations present
orientPrimers <- lapply(c(fwdPrimer, revPrimer), function(primer) {
  dna <- DNAString(primer)
  orients <- c(Forward = dna, Complement = Biostrings::complement(dna),
               Reverse = reverse(dna), RevComp = reverseComplement(dna))
  return(sapply(orients, toString))
})

names(orientPrimers) <- c("fwdPrimer", "revPrimer")

write.table(as.matrix(unlist(orientPrimers)), file = "../results/orientPrimers.txt",
```

```
            col.names = F, sep = "\t", quote = F)

# Counts number of reads containing the primer sequence
primerHits <- function(primer, fn, max.mismatch = 0) {
  nhits <- vcountPattern(primer, sread(readFastq(fn)), fixed = FALSE)
  return(sum(nhits > 0))
}

loc <- "../results/pmHits/"
dir.create(loc, showWarnings = F)

fls <- list.files("../results/filtFASTQ/", pattern = ".gz", full.names = T)
idx <- tools::file_path_sans_ext(list.files(loc))
fls <- fls[!gsub("\\.fastq.gz", "", basename(fls)) %in% idx]

if (length(fls) != 0) {
  registerDoFuture()
  options(future.globals.maxSize = 20*1024^3)
  plan(multiprocess, workers = availableCores() - 1)

  foreach(i = 1:length(fls)) %dopar% {
    x <- sapply(unlist(orientPrimers), primerHits, fn = fls[i], max.mismatch = 5)
    saveRDS(x, file = paste0(loc, gsub("\\.fastq.gz", "", basename(fls[i])), ".Rds"))
  }

  plan(sequential)
}

tbl <- sapply(list.files("../results/pmHits/", full.names = T), readRDS)
tbl <- t(tbl)
rownames(tbl) <- paste0(tools::file_path_sans_ext(basename(rownames(tbl))), ".fastq.gz")
ord <- gsub(".fastq.gz", "", rownames(tbl))
ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
ord <- order(ord)
tbl <- tbl[ord,]

kable(format(tbl, big.mark = ","), format = 'html', align = "r",
      col.names = sapply(strsplit(colnames(tbl), ".", fixed = T), "[[", 2),
      caption = "Frequency of the primer sequences in various orientations in the swapped paired-end read") %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  column_spec(1, bold = T) %>%
  add_header_above(c(" " = 1, "Forward Primer" = 4, "Reverse Primer" = 4)) %>%
  cat(., file = "../Supplementary Materials/S8.html")
```

PE reads were stitched together using NGmerge at a minimum overlap of 20 nt allowing for a maximum of 25% mismatched. PE reads with ambiguous bases were removed and reads that successfully merged were moved into it own file marked by *_.gz while reads that failed to merge were placed in files marked by *_1.fastq.gz and *_2.fastq.gz for the the forward and reverse reads respectively. Only the merged reads and failed-to-merge forward reads were carried over for further processing.

```
loc <- "../results/filtFASTQ/"

dir.create(paste0(loc, "mergedReads/"), showWarnings = F)
dir.create(paste0(loc, "failedStitch/"), showWarnings = F)

fls <- list.files(loc, pattern = ".gz")
newNames <- unique(gsub("_R.+", "", fls))

fls <- list.files(paste0(loc, c("mergedReads", "failedStitch")))
fls <- gsub("(>?_[12])*(>?\\.fastq)*.gz", "", fls)
idx <- table(fls)
idx <- names(idx[idx == 3])

fwdReads <- list.files(loc, pattern = "_R1.fastq.gz", full.names = T)
names(fwdReads) <- gsub("_R1.fastq.gz", "", basename(fwdReads))
revReads <- list.files(loc, pattern = "_R2.fastq.gz", full.names = T)
```

```
names(revReads) <- gsub("_R2.fastq.gz", "", basename(revReads))

fwdReads <- fwdReads[!names(fwdReads) %in% idx]
revReads <- revReads[!names(revReads) %in% idx]
newNames <- newNames[!newNames %in% idx]

for (i in newNames) {
  system(paste('bash -c',
               shQuote(paste('/mnt/c/NGmerge-master/NGmerge',
                             '-1', shQuote(fwdReads[i]), '-2', shQuote(revReads[i]),
                             '-o', shQuote(paste0(loc, "mergedReads/", i)),
                             '-f', shQuote(paste0(loc, "failedStitch/", i)), '-p', 0.25)), "'"))
}
```

NGS samples were filtered further to facilitate DADA2 error modeling All reads with ambiguous bases (Ns) were removed and further filtered out reads that have bases with quality scores below 10 and exceeded a cumulative expected error of 3 for merged reads or 4 for the forward reads. Additionally, it was decided that reads need to be at least 240 nt long. The filtering parameters were chosen as such to refine error modeling for the DADA2 algorithm downstream of the analysis by improving read quality while retaining as many reads as possible.

```
# Remove PE reads with ambiguous bases (Ns) and perform maxEE filtering
loc <- "../results/filterAndTrim/"
dir.create(loc, showWarnings = F)

fls <- data.frame(merged = list.files("../results/filtFASTQ/mergedReads/",
                                      pattern = ".gz", full.names = T),
                  fwd = list.files("../results/filtFASTQ/failedStitch/",
                                   pattern = "_1.fastq.gz", full.names = T))
idx <- tools::file_path_sans_ext(list.files(loc))
idx <- tools::file_path_sans_ext(basename(fls$merged)) %in% idx
fls <- fls[!idx,]

if (nrow(fls) != 0) {
  registerDoFuture()
  options(future.globals.maxSize = 20*1024^3, future.seed = 1)
  plan(multiprocess, workers = availableCores() - 1)

  foreach(i = 1:nrow(fls)) %dorng% {
    fls.filt <- lapply(fls[i,], function(x) file.path("../results", "filtNminQmaxEEminL", basename(x)))
    names(fls.filt) <- c("merged", "fwd")

    out.merged1 <- filterAndTrim(fwd = fls[i, "merged"], filt = fls.filt$merged, maxN = 0)
    out.merged2 <- filterAndTrim(fwd = fls[i, "merged"], filt = fls.filt$merged, maxN = 0, minQ = 10)
    out.merged3 <- filterAndTrim(fwd = fls[i, "merged"], filt = fls.filt$merged,
                                 maxN = 0, minQ = 10, maxEE = 3)
    out.merged <- filterAndTrim(fwd = fls$merged[i], filt = fls.filt$merged,
                                maxN = 0, minQ = 10, maxEE = 3, minLen = 240)
    out.fwd1 <- filterAndTrim(fwd = fls[i, "fwd"], filt = fls.filt$fwd, maxN = 0)
    out.fwd2 <- filterAndTrim(fwd = fls[i, "fwd"], filt = fls.filt$fwd, maxN = 0, minQ = 10)
    out.fwd3 <- filterAndTrim(fwd = fls[i, "fwd"], filt = fls.filt$fwd,
                              maxN = 0, minQ = 10, maxEE = 4)
    out.fwd <- filterAndTrim(fwd = fls[i, "fwd"], filt = fls.filt$fwd,
                             maxN = 0, minQ = 10, maxEE = 4, minLen = 240)

    out <- rbind(cbind(out.merged1, out.merged2[,2], out.merged3[,2], out.merged[,2]),
                 cbind(out.fwd1, out.fwd2[,2], out.fwd3[,2], out.fwd[,2]))
    colnames(out) <- c("read.in", "reads.out.N", "reads.out.NminQ", "reads.out.NminQmaxEE",
                       "reads.out.NminQmaxEEminL")

    saveRDS(out, file = paste0(loc, tools::file_path_sans_ext(basename(fls$merged[i])), ".Rds"))
  }

  plan(sequential)
}
```

```
# Get file paths to sequences
fls <- list.files("../results/filtNminQmaxEEminL/", pattern = ".gz", full.names = T)
ord <- mgsub::mgsub(basename(fls), c("^.*(H\\d+_R[12]).+", "(\\d+[A-Z]_R[12]).+"), c("\\1", "\\1"))
ord <- gsub("\\..+", "", ord)
ids <- gsub("_1", "", ord)
ids <- unique(ids)

# Reexamine quality scores after NGmerge
loc <- "../results/fig/plotQualityProfilefiltNmaxEEminL/"
dir.create(loc, showWarnings = F)
for (i in ids) {
  fl.loc <- paste0(loc, "plotQualityProfilefiltNmaxEEminL-", i, ".png")
  if (!file.exists(fl.loc)) {
    png(fl.loc, height = 3, width = 8, res = 300, units = "in")
    print(plotQualityProfile(grep(paste0(i, "(?>_1)*\\."), fls, value = T, perl = T), n = 2E6) +
            facet_wrap(~file, ncol = 2))
    dev.off()
  }
}

zip("../Supplementary Materials/S9.zip",
    files = list.files("../results/fig/plotQualityProfilefiltNmaxEEminL/", full.names = T),
    include_directories = F, mode = "cherry-pick")

# Tabulate results
tbl <- lapply(list.files("../results/filterAndTrim/", full.names = T), readRDS)
tbl <- do.call(rbind, tbl)
tbl <- data.frame(tbl, percent.retained = round(100*tbl[,ncol(tbl)]/tbl[,1], 1))
ord <- gsub("(_[12]\\.fastq)*\\.gz", "", rownames(tbl))
ord <- mdat[match(gsub("_R[12]", "", ord), mdat$Code), "AltCode"]
ord <- order(ord)
tbl <- tbl[ord,]

kable(format(tbl, big.mark = ","), format = 'html', align = "r",
      col.names = c("reads in", "after N filtering", "after minQ filtering",
                    "after maxEE filtering", "after minLen filtering", "percent retained"),
      caption = "Number of reads before and after filtering with percent of reads remaining") %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  column_spec(1, bold = T) %>%
  add_header_above(c(" ", " ", "reads out" = 4, "")) %>%
  cat(., file = "../Supplementary Materials/S10.html")
```

The quality profiles of the filtered reads can be found in **S9.zip**. The breakdown of the reads retained is given in **S10.html**, as it passed through the four different rounds of filtering (N removal, minimum quality of 10, maximum cumulative error of 3 or 4 for merged and forward-only respectively, and minimum length of 240 nt).

The filtered reads are tallied by unique reads to speed up downstream calculations.

```
# Tally unique sequences
fls <- list(merged = list.files("../results/filtNminQmaxEEminL/", pattern = "[^q].gz",
                                full.names = T),
            fwd = list.files("../results/filtNminQmaxEEminL/", pattern = "_1.fastq.gz",
                             full.names = T))
dereps <- lapply(fls, derepFastq, verbose = T)
```

Then, an error model is estimated to distinguish sequencing error from biological variation.

```
# Model substitution errors
ptm <- proc.time()
plan(multiprocess, workers = availableCores() - 1)
errs <- future_lapply(dereps, learnErrors, multithread = T, nbases = 1E10, future.seed = T)
plan(sequential)
proc.time() - ptm  # run time: > 4 h

my_save(errs, file = "../results/errs.Rds", overwrite = T)
```

```
for (i in names(errs)) {
  ggsave(paste0(i, ".png"), plotErrors(errs[[i]], nominalQ = T) + labs(title = i), device = "png",
         path = "../results/fig/", width = 12, height = 12, units = "in", dpi = 150)
}


zip("../Supplementary Materials/S11.zip",
    files = list.files("../results/fig/", pattern = "fwd|merged", full.names = T),
    include_directories = F, mode = "cherry-pick")
```

Model fit seems reasonably good to use in DADA2's algorithm to infer ASVs (amplicon sequence variants) except at low consensus quality scores for some nucleotide substitutions (**S11.zip**).

```
# Perform DADA2 sequence inference
errs <- readRDS("../results/errs.Rds")

ptm <- proc.time()
plan(multiprocess, workers = availableCores() - 1)
dadas <- future_Map(dada, derep = dereps, err = errs,
                    MoreArgs = list(pool = T, multithread = T), future.seed = T)
plan(sequential)
proc.time() - ptm  # run time: > 6 h

my_save(dadas, compress = F, file = "../results/dadas.Rds", overwrite = F)

# Merge
dadas <- readRDS("../results/dadas.Rds")

seqtabAll <- makeSequenceTable(unlist(dadas, recursive = F))

seqtabNoC <- removeBimeraDenovo(seqtabAll)

seqtabNoC <- seqtabNoC[,order(nchar(colnames(seqtabNoC)))]

my_save(seqtabAll, file = "../results/seqtabAll.Rds", overwrite = F)
my_save(seqtabNoC, file = "../results/seqtabNoC.Rds", overwrite = F)
```

The number of ASVs found was 9,645. This number decreased to 7,613 after filtering for chimeric sequences, the results of which are tabulated below.

A table of ASV counts across all samples can be found in **S12.csv** while a table of ASV length distributions is given in **S13.html**.

Taxonomic assignment via BLAST

Most ASVs matched to ITS2 sequences from flowering plants but a few remain unidentified
Sequences were BLASTed to identify the source organism. Results were restricted to the top ten sequences producing significant alignments and limited to records that include **Viridiplantae (taxid:33090)**. Please refer to BLAST manual for the description of the column headers of the table below.

```
# Write ASVs to FASTA
seqtabNoC <- readRDS("../results/seqtabNoC.Rds")
seqs <- getSequences(seqtabNoC)
seqs <- DNAStringSet(seqs)
names(seqs) <- paste0("Query_", seq_along(seqs))

writeXStringSet(seqs, "../results/seqs.fa")

## Run remote BLAST (from a terminal); can take a long time to finish
time blastn -db nt -query ./results/seqs.fa -outfmt "7 qseqid sseqid pident qlen slen length mismatch \
gapopen qstart qend sstart send evalue bitscore sscinames salltitles" -max_target_seqs 10 -max_hsps 10 -out \
 .results/seqs.out -remote -entrez_query  "Viridiplantae"[Organism]' -strand 'plus'
```

```
tbl <- read.table("../results/seqs.out", sep = "\t", fill = T, quote = "", stringsAsFactors = T)

# Tabulate the tally of ASV counts
cnames <- readLines("../results/seqs.out", n = 5)[5]
cnames <- unlist(strsplit(cnames, ", "))
cnames <- gsub("^.+: ", "", cnames)
colnames(tbl) <- cnames

write.csv(tbl, file = "../Supplementary Materials/S14.csv")
```

BLAST results restricted to Viridiplantae sequences can be found in **S14.csv**.

Of the 7,613 ASVs, 7,590 ASVs aligned to Viridiplantae ITS2 sequences. Percent identity varied from 73.8% to 100% for the top matches, where 2,047 out of 7,590 have 100% identity.

```
# Write ASVs that didn't match to FASTA
seqs <- seqs[as.numeric(gsub("Query_", "", setdiff(names(seqs), levels(tbl[[1]]))))]

writeXStringSet(seqs, "../results/seqsNoMatch.fa")

## Run remote BLAST (from a terminal); finishes quickly due to a smaller number of reads to process
time blastn -db nt -query ./results/seqsNoMatch.fa -outfmt "7 qseqid sseqid pident qlen slen length mismatch \
gapopen qstart qend sstart send evalue bitscore sscinames salltitles" -max_target_seqs 10 -max_hsps 10 -out \
./results/seqsNoMatch.out -remote -strand 'plus'

tbl <- read.table("../results/seqsNoMatch.out", sep = "\t", fill = T, quote = "\"")

# Tabulate the tally of ASV counts
cnames <- readLines("../results/seqs.out", n = 5)[5]
cnames <- unlist(strsplit(cnames, ", "))
cnames <- gsub("^.+: ", "", cnames)
colnames(tbl) <- cnames

write.csv(tbl, file = "../Supplementary Materials/S15.csv")
```

BLAST results that did not match to Viridiplantae ITS2 sequences but did match to a sequence in the NCBI *nt* database can be found in **S15.csv**.

The remaining 23 ASVs were reBLASTed using NCBI's entire *nt* database. It resulted to 4 additional alignments, leaving 19 ASVs unidentified.

```
# Get unmatched ASVs
x <- unlist(read.table("../results/seqsNoMatch.fa", stringsAsFactors = F), use.names = F)
x[grepl(">", x)] <- paste0(x[grepl(">", x)], " ")
x <- paste0(x, collapse = "")
x <- unlist(strsplit(x, ">", fixed = T), recursive = F)
x <- x[nchar(x) > 0]
x <- x[!gsub("(Query_\\d+) .+", "\\1", x) %in% tbl$`query id`]
x <- paste0(">", x)

writeLines(x, file("../Supplementary Materials/S16.txt"))
closeAllConnections()
```

The unmatched ASVs can be found in **S16.txt**.

ASVs with the same taxonomic assignment are aggregated based on alignment and mismatch
Three numbers are appended to each taxonomic assignment, creating a new label. The first number refers to the ASV length, the second number indicates the portion of the ASV length that mapped to an *nt* database sequence, and the third number is the number of mismatches. ASVs with identical labels are aggregated by summing their read counts.

Taxonomic assignments are marked with asterisks when the first nucleotide of the query doesn't match to the subject sequence from the NCBI *nt* database or if the position of the last nucleotide in the query is inconsistent with the query length. These ASVs should be carefully examined as the reliability of the

taxonomic assignments should be considered suspect.

```
# Extract top results from segregated BLAST queries
Qys <- read.table("../results/seqs.out", sep = "\t", stringsAsFactors = F, quote = "")
Qys <- aggregate(Qys, Qys[1], "[[", 1)
Qns <- read.table("../results/seqsNoMatch.out", sep = "\t", stringsAsFactors = F, quote = "")
Qns <- aggregate(Qns, Qns[1], "[[", 1)
topHits <- rbind(Qys, Qns)

# Replace empty BLAST sscinames with info from BLAST salltitles if available
topHits[topHits[[16]] == "N/A", 16] <- gsub("(.+?) (.+?) .+", "\\1 \\2", topHits[topHits[[16]] == "N/A", 17])

# Order queries by number
topHits <- topHits[order(as.numeric(gsub("Query_", "", topHits[[1]]))),]

# Create labels. Add an asterisk if the first nucleotide of the query doesn't match or if the
# position of the last nucleotide in the query is inconsistent with the query length
taxa <- paste0(topHits[[16]], " (", paste(topHits[[5]], topHits[[7]], topHits[[8]], sep = "/"), ")",
               ifelse(topHits[[10]] != 1 | topHits[[11]] != topHits[[5]], "*", " "))

# Aggregate ASVs based on labels (taxonomic assignment + BLAST info)
# Segregate ASVs with no assigned taxa from BLAST
tbl <- readRDS("../results/tbl-ASVs.Rds")
tbl.taxa <- aggregate(tbl[gsub("(Query_\\d+).+", "\\1", rownames(tbl)) %in% topHits[[1]],],
                      by = list(taxa = taxa), sum)
rownames(tbl.taxa) <- tbl.taxa$taxa
tbl.taxa <- tbl.taxa[,-1]
tbl.unkn <- tbl[!gsub("(Query_\\d+).+", "\\1", rownames(tbl)) %in% topHits[[1]],]
rownames(tbl.unkn) <- gsub("(Query_\\d+).*", "\\1", rownames(tbl.unkn))

# Get total counts per label
mat <- rowSums(tbl.taxa)
mat <- cbind(mat, do.call(rbind, strsplit(gsub(".+\\((.+)\\).+$", "\\1", rownames(tbl.taxa)), "/")))
colnames(mat) <- c("totalCts", "ASV_length", "mapped_length", "mismatch")
class(mat) <- "numeric"


write.csv(tbl.taxa, file = "../Supplementary Materials/S17.csv")
write.csv(tbl.unkn, file = "../Supplementary Materials/S18.csv")
write.csv(mat, file = "../Supplementary Materials/S19.csv")
```

The aggregated ASVs with and without taxonomic assignments are tabulated in **S17.csv** and **S18.csv** respectively. The total count for each label across all samples is given in **S19.csv**.

Analysis of Results

t-SNE

Samples that didn't pass the mismatch criteria (20 or less mismatches) were excluded. Additionally, samples with an average of 2 or less read counts per ASV present were also taken out. Read counts were expressed as fractions of the total number of reads per sample (ASV relative count). Samples are split between merged and forward-only sequences.

Clustering was performed using t-distributed stochastic neighbor embedding (t-SNE) using the exact algorithm. A perplexity of 3 was chosen to reflect the presence of replicate sample runs in the dataset. Closeness of points on the plot indicates similarity of results based on ASV relative counts. Samples are color-coded based on geographic location, with sample ID and origin indicated (see *S1.csv*) on separate plots.

```
# Extract taxonomic assignments passing the criteria
idx <- mat[,"mismatch"] <= 20

tbl <- tbl.taxa[idx,]
tbl <- tbl[,colSums(tbl) > 0]
tbl <- tbl[,colSums(tbl)/colSums(tbl > 0) > 2]
ptbl <- prop.table(as.matrix(tbl), margin = 2)
```

```
colnames(ptbl) <- gsub("_1.fastq", "", tools::file_path_sans_ext(colnames(ptbl)))
colnames(ptbl) <- gsub("^(.).+(\\..+)", "\\1\\2", colnames(ptbl))

ptbl.m <- ptbl[,grepl("^m", colnames(ptbl))]
ptbl.f <- ptbl[,grepl("^f", colnames(ptbl))]

id <- as.factor(mdat[match(gsub("^.+\\.", "", colnames(ptbl)), mdat$Code), "Continent"])
id <- as.factor(gsub("[Nn]orth|[Ss]outh|[Ee]ast|[Ww]est|[Cc]entral|and| ", "", id))
names(id) <- colnames(ptbl)
```

- t-SNE plots of merged reads can be found in **S20.png**.

```
# Plot t-SNE results of filtered merged reads
set.seed(2021)
clr.m <- adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha = 0.7)[id[grepl("^m\\.", names(id))]]
res.m <- Rtsne(t(ptbl.m), perplexity = 3, theta = 0.0)
axs.m <- c(range(res.m$Y[,1]), range(res.m$Y[,2]))*1.2

png("../Supplementary Materials/S20.png", width = 20, height = 20, units = "in", res = 300)
layout(mat = matrix(c(1, 1, 2, 3, 4, 5), nrow = 3, ncol = 2, byrow = T), heights = c(0.04, 0.48, 0.48))
par(mar = c(0, 0, 0, 0))
plot(1, type = "n", xlab = "", ylab = "", ann = F, axes = F)
legend("center", legend = levels(id), col = adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha.f = 0.7),
       pch = 15, pt.cex = 2.5, ncol = 1, bty = "n", xpd = T, horiz = T)
par(mar = c(0.2, 0.2, 0.2, 0.2) + 0.1)
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
wordcloud::textplot(res.m$Y[,1], res.m$Y[,2], main = "", xlab = "", ylab = "", words = colnames(ptbl.m), new = F,
                    xlim = axs.m[1:2], ylim = axs.m[3:4])
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
country <- mdat$Country[match(gsub("^m.", "", colnames(ptbl.m)), mdat$Code)]
wordcloud::textplot(res.m$Y[,1], res.m$Y[,2], main = "", xlab = "", ylab = "", words = country, new = F,
                    xlim = axs.m[1:2], ylim = axs.m[3:4])
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
region <- mdat$Continent[match(gsub("^m.", "", colnames(ptbl.m)), mdat$Code)]
region <- mgsub::mgsub(region, c("North", "South", "East", "West", "Southeast"), c("N", "S", "E", "W", "SE"))
wordcloud::textplot(res.m$Y[,1], res.m$Y[,2], main = "", xlab = "", ylab = "", words = region, new = F,
                    xlim = axs.m[1:2], ylim = axs.m[3:4])
dev.off()
```

- t-SNE plots of forward-only reads can be found in **S21.png**.

```
# Plot t-SNE results of filtered forward-only reads
set.seed(2021)
clr.f <- adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha = 0.7)[id[grepl("^f\\.", names(id))]]
res.f <- Rtsne(t(ptbl.f), perplexity = 3, theta = 0.0)
axs.f <- c(range(res.f$Y[,1]), range(res.f$Y[,2]))*1.2

png("../Supplementary Materials/S21.png", width = 20, height = 20, units = "in", res = 300)
layout(mat = matrix(c(1, 1, 2, 3, 4, 5), nrow = 3, ncol = 2, byrow = T), heights = c(0.04, 0.48, 0.48))
par(mar = c(0, 0, 0, 0))
plot(1, type = "n", xlab = "", ylab = "", ann = F, axes = F)
legend("center", legend = levels(id), col = adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha.f = 0.7),
       pch = 15, pt.cex = 2.5, ncol = 1, bty = "n", xpd = T, horiz = T)
par(mar = c(0.2, 0.2, 0.2, 0.2) + 0.1)
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
wordcloud::textplot(res.f$Y[,1], res.f$Y[,2], main = "", xlab = "", ylab = "", words = colnames(ptbl.f), new = F,
                    xlim = axs.f[1:2], ylim = axs.f[3:4])
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
country <- mdat$Country[match(gsub("^f.", "", colnames(ptbl.f)), mdat$Code)]
wordcloud::textplot(res.f$Y[,1], res.f$Y[,2], main = "", xlab = "", ylab = "", words = country, new = F,
                    xlim = axs.f[1:2], ylim = axs.f[3:4])
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
region <- mdat$Continent[match(gsub("^f.", "", colnames(ptbl.f)), mdat$Code)]
region <- mgsub::mgsub(region, c("North", "South", "East", "West", "Southeast"), c("N", "S", "E", "W", "SE"))
wordcloud::textplot(res.f$Y[,1], res.f$Y[,2], main = "", xlab = "", ylab = "", words = region, new = F,
```

```
                  xlim = axs.f[1:2], ylim = axs.f[3:4])
dev.off()
```

ASVs and samples were filtered using blast hit information and the visualization is repeated using a curated set of queries and samples.

```
# Extract taxonomic assignments based on curated list
query <- unlist((read.table("../results/goodASVs")), use.names = F)
samples <- unlist(read.table("../results/goodSamples"), use.names = F)

tbl <- readRDS("../results/tbl-ASVs.Rds")
ord <- gsub("_1.fastq", "", colnames(tbl))
ord <- gsub("^(.).*\\.(.+)\\..+", "\\1-\\2", ord)
ord <- gsub("H(\\d)\\b", "H0\\1", ord)
ord <- gsub("H15$", "H15-0", ord, perl = T)
ord <- gsub("-25", "-H15", ord, perl = T)
ord <- mgsub::mgsub(ord, c("^m", "^f"), c(1, 2))
ord <- order(ord)
colnames(tbl) <- gsub(".gz|_1.fastq.gz", "", colnames(tbl))
tbl <- tbl[gsub("(Query_\\d+) .*", "\\1", rownames(tbl)) %in% query,
           gsub(".+\\.", "", colnames(tbl)) %in% samples]
tbl <- tbl[,colSums(tbl) > 0]

ptbl <- prop.table(as.matrix(tbl), margin = 2)
colnames(ptbl) <- gsub("^(.).+(\\..+)", "\\1\\2", colnames(ptbl))

ptbl.m <- ptbl[,grepl("^m", colnames(ptbl))]
ptbl.f <- ptbl[,grepl("^f", colnames(ptbl))]
id <- as.factor(mdat[match(gsub("^.+\\.", "", colnames(ptbl)), mdat$Code), "Continent"])
id <- as.factor(gsub("[Nn]orth|[Ss]outh|[Ee]ast|[Ww]est|[Cc]entral|and| ", "", id))
names(id) <- colnames(ptbl)
```

- t-SNE plots of curated merged reads can be found in **S22.png**.

```
# Plot t-SNE results of curated merged reads
set.seed(2021)
clr.m <- adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha = 0.7)[id[grepl("^m\\.", names(id))]]
res.m <- Rtsne(t(ptbl.m), perplexity = 5, theta = 0.0, check_duplicates = F)
axs.m <- c(range(res.m$Y[,1]), range(res.m$Y[,2]))*1.2

png("../Supplementary Materials/S22.png", width = 20, height = 20, units = "in", res = 300)
layout(mat = matrix(c(1, 1, 2, 3, 4, 5), nrow = 3, ncol = 2, byrow = T), heights = c(0.04, 0.48, 0.48))
par(mar = c(0, 0, 0, 0))
plot(1, type = "n", xlab = "", ylab = "", ann = F, axes = F)
legend("center", legend = levels(id), col = adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha.f = 0.7),
       pch = 15, pt.cex = 2.5, ncol = 1, bty = "n", xpd = T, horiz = T)
par(mar = c(0.2, 0.2, 0.2, 0.2) + 0.1)
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
wordcloud::textplot(res.m$Y[,1], res.m$Y[,2], main = "", xlab = "", ylab = "", words = colnames(ptbl.m), new = F,
                    xlim = axs.m[1:2], ylim = axs.m[3:4])
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
country <- mdat$Country[match(gsub("^m.", "", colnames(ptbl.m)), mdat$Code)]
wordcloud::textplot(res.m$Y[,1], res.m$Y[,2], main = "", xlab = "", ylab = "", words = country, new = F,
                    xlim = axs.m[1:2], ylim = axs.m[3:4])
plot(res.m$Y, col = clr.m, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.m[1:2], ylim = axs.m[3:4])
region <- mdat$Continent[match(gsub("^m.", "", colnames(ptbl.m)), mdat$Code)]
region <- mgsub::mgsub(region, c("North", "South", "East", "West", "Southeast"), c("N", "S", "E", "W", "SE"))
wordcloud::textplot(res.m$Y[,1], res.m$Y[,2], main = "", xlab = "", ylab = "", words = region, new = F,
                    xlim = axs.m[1:2], ylim = axs.m[3:4])
dev.off()
```

- t-SNE plots of curated forward-only reads can be found in **S23.png**.

```
# Plot t-SNE results of curated forward-only reads
set.seed(2021)
clr.f <- adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha = 0.7)[id[grepl("^f\\.", names(id))]]
```

```
res.f <- Rtsne(t(ptbl.f), perplexity = 3, theta = 0.0, check_duplicates = F)
axs.f <- c(range(res.f$Y[,1]), range(res.f$Y[,2]))*1.2

png("../Supplementary Materials/S23.png", width = 20, height = 20, units = "in", res = 300)
layout(mat = matrix(c(1, 1, 2, 3, 4, 5), nrow = 3, ncol = 2, byrow = T), heights = c(0.04, 0.48, 0.48))
par(mar = c(0, 0, 0, 0))
plot(1, type = "n", xlab = "", ylab = "", ann = F, axes = F)
legend("center", legend = levels(id), col = adjustcolor(brewer.pal(length(levels(id)), "Set1"), alpha.f = 0.7),
       pch = 15, pt.cex = 2.5, ncol = 1, bty = "n", xpd = T, horiz = T)
par(mar = c(0.2, 0.2, 0.2, 0.2) + 0.1)
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
wordcloud::textplot(res.f$Y[,1], res.f$Y[,2], main = "", xlab = "", ylab = "", words = colnames(ptbl.f), new = F,
                    xlim = axs.f[1:2], ylim = axs.f[3:4])
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
country <- mdat$Country[match(gsub("^f.", "", colnames(ptbl.f)), mdat$Code)]
wordcloud::textplot(res.f$Y[,1], res.f$Y[,2], main = "", xlab = "", ylab = "", words = country, new = F,
                    xlim = axs.f[1:2], ylim = axs.f[3:4])
plot(res.f$Y, col = clr.f, pch = 16, xaxt = "n", yaxt = "n", ann = F, cex = 1.5, xlim = axs.f[1:2], ylim = axs.f[3:4])
region <- mdat$Continent[match(gsub("^f.", "", colnames(ptbl.f)), mdat$Code)]
region <- mgsub::mgsub(region, c("North", "South", "East", "West", "Southeast"), c("N", "S", "E", "W", "SE"))
wordcloud::textplot(res.f$Y[,1], res.f$Y[,2], main = "", xlab = "", ylab = "", words = region, new = F,
                    xlim = axs.f[1:2], ylim = axs.f[3:4])
dev.off()
```

R packages used

Aboyoun, P., H. Pagès, and M. Lawrence. 2020. *GenomicRanges: Representation and Manipulation of Genomic Intervals*. https://bioconductor.org/packages/GenomicRanges.

Ahlmann-Eltze, Constantin, Peter Hickey, and Hervé Pagès. 2021. *MatrixGenerics: S4 Generic Summary Statistic Functions That Operate on Matrix-Like Objects*. https://bioconductor.org/packages/MatrixGenerics.

Arora, Sonali, Martin Morgan, Marc Carlson, and H. Pagès. 2021. *GenomeInfoDb: Utilities for Manipulating Chromosome Names, Including Modifying Them to Follow a Particular Naming Style*. https://bioconductor.org/packages/GenomeInfoDb.

Barter, Rebecca, and Bin Yu. 2021. *Superheat: A Graphical Tool for Exploring Complex Datasets Using Heatmaps*.

Bengtsson, Henrik. 2020a. *Future: Unified Parallel and Distributed Processing in r for Everyone*. https://github.com/HenrikBengtsson/future.

———. 2020b. "A Unifying Framework for Parallel and Distributed Processing in r Using Futures." https://arxiv.org/abs/2008.00553.

———. 2020d. "A Unifying Framework for Parallel and Distributed Processing in r Using Futures." https://arxiv.org/abs/2008.00553.

———. 2020c. "A Unifying Framework for Parallel and Distributed Processing in r Using Futures." https://arxiv.org/abs/2008.00553.

———. 2021a. *doFuture: A Universal Foreach Parallel Adapter Using the Future API of the Future Package*. https://github.com/HenrikBengtsson/doFuture.

———. 2021b. *Future.apply: Apply Function to Elements in Parallel Using Futures*. https://github.com/HenrikBengtsson/future.apply.

———. 2021c. *matrixStats: Functions That Apply to Rows and Columns of Matrices (and to Vectors)*. https://github.com/HenrikBengtsson/matrixStats.

Callahan, Benjamin J, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. 2016. "Dada2: High-Resolution Sample Inference from Illumina Amplicon Data." *Nature*

Methods 13: 581–83. https://doi.org/10.1038/nmeth.3869.

Callahan, Benjamin, Paul McMurdie, and Susan Holmes. 2020. *Dada2: Accurate, High-Resolution Sample Inference from Amplicon Sequencing Data*. http://benjjneb.github.io/dada2/.

Cheng, Joe, Carson Sievert, Winston Chang, Yihui Xie, and Jeff Allen. 2021. *Htmltools: Tools for HTML*. https://github.com/rstudio/htmltools.

Csárdi, Gábor, Kuba Podgórski, and Rich Geldreich. 2020. *Zip: Cross-Platform Zip Compression*. https://github.com/r-lib/zip#readme.

Eddelbuettel, Dirk. 2013. *Seamless R and C++ Integration with Rcpp*. New York: Springer. https://doi.org/10.1007/978-1-4614-6868-4.

Eddelbuettel, Dirk, and James Joseph Balamuta. 2018. "Extending extitR with extitC++: A Brief Introduction to extitRcpp." *The American Statistician* 72 (1): 28–36. https://doi.org/10.1080/00031305.2017.1375990.

Eddelbuettel, Dirk, Romain Francois, JJ Allaire, Kevin Ushey, Qiang Kou, Nathan Russell, Douglas Bates, and John Chambers. 2021. *Rcpp: Seamless r and c++ Integration*. https://CRAN.R-project.org/package=Rcpp.

Eddelbuettel, Dirk, and Romain François. 2011. "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software* 40 (8): 1–18. https://doi.org/10.18637/jss.v040.i08.

Ewing, Mark. 2020. *Mgsub: Safe, Multiple, Simultaneous String Substitution*. https://CRAN.R-project.org/package=mgsub.

Fellows, Ian. 2018. *Wordcloud: Word Clouds*. https://CRAN.R-project.org/package=wordcloud.

Galili, Tal. 2019. *Installr: Using r to Install Stuff on Windows OS (Such as: R, Rtools, RStudio, Git, and More!)*. https://CRAN.R-project.org/package=installr.

Gaujoux, Renaud. 2020a. *doRNG: Generic Reproducible Parallel Backend for Foreach Loops*. https://renozao.github.io/doRNG.

———. 2020b. *Rngtools: Utility Functions for Working with Random Number Generators*. https://renozao.github.io/rngtools.

Gentleman, R., V. Carey, M. Morgan, and S. Falcon. 2020. *Biobase: Base Functions for Bioconductor*. https://bioconductor.org/packages/Biobase.

Huber, W., V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, et al. 2015a. "Orchestrating High-Throughput Genomic Analysis with Bioconductor." *Nature Methods* 12 (2): 115–21. http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

Huber, W., Carey, V. J., Gentleman, R., Anders, et al. 2015b. "Orchestrating High-Throughput Genomic Analysis with Bioconductor." *Nature Methods* 12 (2): 115–21. http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

Krijthe, Jesse. 2018. *Rtsne: T-Distributed Stochastic Neighbor Embedding Using a Barnes-Hut Implementation*. https://github.com/jkrijthe/Rtsne.

Lawrence, Michael, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin Morgan, and Vincent Carey. 2013a. "Software for Computing and Annotating Genomic Ranges." *PLoS Computational Biology* 9. https://doi.org/10.1371/journal.pcbi.1003118.

———. 2013b. "Software for Computing and Annotating Genomic Ranges." *PLoS Computational Biology* 9. https://doi.org/10.1371/journal.pcbi.1003118.

———. 2013c. "Software for Computing and Annotating Genomic Ranges." *PLoS Computational Biology* 9. https://doi.org/10.1371/journal.pcbi.1003118.

Morgan, Martin, Simon Anders, Michael Lawrence, Patrick Aboyoun, Hervé Pagès, and Robert Gentleman. 2009. "ShortRead: A Bioconductor Package for Input, Quality Assessment and Exploration of High-Throughput Sequence Data." *Bioinformatics* 25: 2607–8. https://doi.org/10.1093/bioinformatics/btp450.

Morgan, Martin, Michael Lawrence, and Simon Anders. 2020. *ShortRead: FASTQ Input and Manipulation*.

Morgan, Martin, Valerie Obenchain, Jim Hester, and Hervé Pagès. 2020. *SummarizedExperiment: SummarizedExperiment Container*. https://bioconductor.org/packages/SummarizedExperiment.

Morgan, Martin, Valerie Obenchain, Michel Lang, Ryan Thompson, and Nitesh Turaga. 2020. *BiocParallel: Bioconductor Facilities for Parallel Evaluation*. https://github.com/Bioconductor/BiocParallel.

Morgan, Martin, Hervé Pagès, Valerie Obenchain, and Nathaniel Hayden. 2020. *Rsamtools: Binary Alignment (BAM), FASTA, Variant Call (BCF), and Tabix File Import*. https://bioconductor.org/packages/Rsamtools.

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Pagès, H., P. Aboyoun, R. Gentleman, and S. DebRoy. 2020. *Biostrings: Efficient Manipulation of Biological Strings*. https://bioconductor.org/packages/Biostrings.

Pagès, H., P. Aboyoun, and M. Lawrence. 2020. *IRanges: Foundation of Integer Range Manipulation in Bioconductor*. https://bioconductor.org/packages/IRanges.

Pagès, H., M. Lawrence, and P. Aboyoun. 2020. *S4Vectors: Foundation of Vector-Like and List-Like Containers in Bioconductor*. https://bioconductor.org/packages/S4Vectors.

Pagès, Hervé, and Patrick Aboyoun. 2020. *XVector: Foundation of External Vector Representation and Manipulation in Bioconductor*. https://bioconductor.org/packages/XVector.

Pagès, Hervé, Valerie Obenchain, and Martin Morgan. 2020. *GenomicAlignments: Representation and Manipulation of Short Genomic Alignments*. https://bioconductor.org/packages/GenomicAlignments.

Revolution Analytics, and Steve Weston. n.d. *Foreach: Provides Foreach Looping Construct*.

Rinker, Tyler W., and Dason Kurkiewicz. 2018. *pacman: Package Management for R*. Buffalo, New York. http://github.com/trinker/pacman.

Rinker, Tyler, and Dason Kurkiewicz. 2019. *Pacman: Package Management Tool*. https://github.com/trinker/pacman.

Solymos, Peter, and Zygmunt Zawadzki. 2020. *Pbapply: Adding Progress Bar to *Apply Functions*. https://github.com/psolymos/pbapply.

Team, The Bioconductor Dev. 2021. *BiocGenerics: S4 Generic Functions Used in Bioconductor*. https://bioconductor.org/packages/BiocGenerics.

Vaidyanathan, Ramnath, Yihui Xie, JJ Allaire, Joe Cheng, Carson Sievert, and Kenton Russell. 2020. *Htmlwidgets: HTML Widgets for r*. https://github.com/ramnathv/htmlwidgets.

van der Maaten, L. J. P. 2014. "Accelerating t-SNE Using Tree-Based Algorithms." *Journal of Machine Learning Research* 15: 3221–45.

van der Maaten, L. J. P., and G. E. Hinton. 2008. "Visualizing High-Dimensional Data Using t-SNE." *Journal of Machine Learning Research* 9: 2579–2605.

Wickham, Hadley. 2019. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, and Jennifer Bryan. 2019. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in R." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.

———. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. https://yihui.org/knitr/.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

Zhu, Hao. 2021. *kableExtra: Construct Complex Table with Kable and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.

Blast result based filtering

ASVs were searched against NCBI's nucleotide database (blastn) to identify the source organism. Results were restricted to the top ten sequences producing significant alignments and limited to records that include Viridiplantae (taxid:33090).

Each ASV is assigned a species based on the top blast hit. ASVs that are short or repetitive usually don't have a good blast hit in the GenBank database. We used this fact to further filter ASVs. ASVs with top blast hits that satisfy at least one of the criteria below are filtered out for low quality:

a) percent identity in the alignment less than 85
b) alignment length less than 1/3 of the length of the ASV (query)
c) alignment is less than 150 bp long
d) blast bit score is less than 200
e) species name includes a match to "environmental sample" or matches 'N/A'

After filtering out low-quality ASVs, we also filtered out samples with low complexity. For each sample we counted the number of ASVs detected in the sample with at least 10 reads. If the sample had less than 3 ASVs that satisfied these criteria it was termed a 'low complexity sample' and removed from analysis.

(code for perl scripts used here in **bold** are below)

parse_blast_all_hits_v4.pl ../seqs_blast.out | grep RESULT:| cut -f2,3,4 > all_results_70pct

```
for i in `cat sampleIDs `; do echo -n $i" ";  sample_to_species.pl $i all_results_70pct
ASV_counts.txt F | grep -v "NO GOOD HITS"| grep -v "Vachellia jacquemontii"| grep -v "NONE"
| awk '{if($1>1){print $0}}'| wc -l; done  >
sampleIDs_plants_clean_counts_2orMoreReads_FWD
```

```
awk '{if($2>2){print $0}}' sampleIDs_plants_clean_counts_2orMoreReads_FWD| cut -d" " -f1 >
goodSamples
```

```
m all_results_70pct | egrep -v  "NO GOOD HITS|Vachellia jacquemontii" |cut -f2> goodASVs
```

parse_blast_all_hits_v4.pl. Perl code:

```
#!/usr/bin/perl -w
# parse blast output to see if the top hit is better than the next one

use strict;

# input: blast output
# output: for each query top hit and is it species unique
# just annotate a bad hit do not remove it
# BAD HIT criteria: $this_al < ($this_ql*0.3) , identity 70%


unless ($ARGV[0]) {
   die "Usage: $0 <blast file>
example:$0 /users/sabo/neurogen/HONEY_DHS/BLAST/seqs_20201120.out \n";
}

open(FILE,$ARGV[0]) or die "Missing file: $ARGV[0] \n";

my $first_line = 1;
my %species_scores = ();
my %species_ident = ();
my $bad_species_score = 'NA';
my $bad_species_ident = 'NA';
my $query_name = '';

LINE:while(<FILE>) {
   my $line = $_; chomp $line;
   #print "$line\n";
   if ($line =~ /^\# BLASTN 2.10.0+/ && $first_line == 0) {
         #print the old results, and reset values
         print "QUERY :: \t" . $query_name . "\n";
         my @scores = sort { $b <=> $a } values %species_scores;
         my $top_score = $scores[0];

         my @top_score_species = ();
         foreach my $species (keys %species_scores) {
            if($species_scores{$species} == $top_score) {
                  push(@top_score_species, $species);
            }
         }

         print "\nSPECIES SCORES for $query_name:\n";
         for my $key ( sort { $species_scores{$b} cmp $species_scores{$a} }
                  keys %species_scores )
```

```
        {
           print "$key - $species_scores{$key}\n";
        }

        if (scalar keys %species_scores == 1){
          print "RESULT:\tNOT species specific, only one species in blast
hits\t$query_name\t$top_score_species[0]\n\n";
        }
        elsif (scalar @top_score_species == 0 ) {
          print "RESULT:\tNO GOOD
HITS\t$query_name\t$bad_species_ident\t$bad_species_score\n\n";
        }
        elsif (scalar @top_score_species > 1) {
           print "RESULT:\tNOT species specific, same scores for multiple
sp.\t$query_name\t$top_score_species[0]\n\n";
        }
        else  {
           #print "RESULT:\tYES species specific
YAY\t$query_name\t$top_score_species[0]\n\n";
           print "RESULT:\tYES species specific
YAY\t$query_name\t$top_score_species[0]\t$species_ident{$top_score_species[0]}\n\n";
        }

        #reset
        %species_scores = ();
        %species_ident = ();
        $bad_species_score = 'NA';
        $bad_species_ident = 'NA';
  }
  elsif ($line =~ /^\# BLASTN 2.10.0+/ && $first_line == 1) {
        $first_line = 0;
  }
  elsif($line !~ /\#/) { # skip header lines
        # continue parsing this query
         my @line = split(/\t/, $line);
         my $this_q = $line[0]; $query_name = $this_q;
         my $this_id = $line[2];
         my $this_ql = $line[3];
         my $this_al = $line[5];
         my $this_bs = $line[13];
         my $this_sp = $line[14];

         # is this a good blast result blast score of 200 is about 1e-50 in evalue
```

```
        if ($this_id <70 || $this_al < ($this_ql*0.3) || $this_al < 150  || $this_bs < 200 ||
$this_sp =~ /environmental sample/|| $this_sp eq 'N/A'
) {
            #print "BAD hit: $line\n";
            if ($bad_species_score eq 'NA') {$bad_species_score = $this_bs;}
            if ($bad_species_ident eq 'NA') {$bad_species_ident = $this_sp;}
            next LINE;
          }
          #print $line . "\n";

          unless (exists $species_scores{$this_sp} ) {$species_scores{$this_sp} = $this_bs;}
          unless (exists $species_ident{$this_sp} ) {$species_ident{$this_sp} = $this_id;}

  }
  else {
        #print "skip:$line\n";
  }

}


close(FILE);

exit;
```

sample_to_species.pl perl code:

```
!/usr/bin/perl -w

use strict;

# input: two files:
#   1) sample of interest
#   2) query - top hit, species specific or not
#   3) ASV count table
#   4) Merged, Forward, or All ASVs counts to use

# output: for that sample list top hit, is it species specific


unless ($ARGV[3]) {
  die "Usage: $0 <list of samples> <query species> <query counts table> <M/F/A>
example: $0  H9 all_results seqs_20201120.asv.sample.table.txt M\n";
```

```
}


## load all the data from argv and files

my $sample = $ARGV[0];
my $sample_merg = 'merged.' . $sample;
my $sample_fwd = 'fwd.' . $sample;

my $subset = $ARGV[3];

open(FILE2,$ARGV[1]) or die "Missing file: $ARGV[1] \n";
my %query_top_hit = ();
while(<FILE2>) {
   my $line = $_; chomp $line;
   my @line = split(/\t/, $line);
   my $sp = $line[2] . "\t" . $line[0];
   $query_top_hit{$line[1]} = $sp;
}
close(FILE2);


open(FILE3,$ARGV[2]) or die "Missing file: $ARGV[2] \n";
my @column_num = ();
while(<FILE3>) {
   my $line = $_; chomp $line;
   if ($line =~ /^query\s+/) { #header
         my @line = split(/\t/, $line);
         my $count = 0;
         foreach (@line){
            my $this_name = $_; $this_name =~ s/\s+//g; # clean up last field windows file mess
end of line
             if ($this_name eq $sample_merg || $this_name eq $sample_fwd )
{push(@column_num, $count)}
            $count++;
         }
         unless (exists($column_num[0]) && exists($column_num[1])) {die "this sample is not in
the ASV table : $sample. Exiting... ";}
   }
   else {
         (my $this_q) = $line =~ /(Query_\d+)\t/;
         my @line = split(/\t/, $line);
         my $total_count = 0;
         if ($subset eq 'A') {
```

```
        $total_count = $line[$column_num[0]] + $line[$column_num[1]];
        }
        elsif ($subset eq 'M'){
           $total_count = $line[$column_num[0]];
        }
        elsif ($subset eq 'F') {
            $total_count = $line[$column_num[1]];
        }
        else {
           die "sorry wrong letter for ASV subset use M/F/A see usage\n";
        }

        if ($total_count > 0 && exists $query_top_hit{$this_q}) {
           print "$total_count\t$this_q\t$query_top_hit{$this_q}\n";
        }
elsif ($total_count > 0){
           print "$total_count\t$this_q\tNONE\n";
        }

   }
}
close(FILE3);

exit;
```

SOP Title: ISOLATION AND EXTRACTION OF POLLEN DNA FROM HONEY			
SOP No. **UH/Honey/01**	**Version No.** **2**	**Effective Date:** **02012020**	**Revision Date:** **07012020**
Written by **Dimple Chavan**	**Reviewed by** **Dr. Katerina Kourentzi**	**Approved by** **Dr. Richard C. Willson**	**Page No.** **Page 1 of 4**

1. PURPOSE

The purpose of this standard operating procedure (SOP) is to describe the steps for isolation of pollen from honey and subsequent extraction of plant genomic DNA from the isolated pollen by Qiagen DNeasy Plant Mini Kit. This SOP also describes the quality control of purified DNA extracts by Nanodrop and Quantifluor for downstream applications like PCR and NGS.

2. SCOPE

The SOP can be applied to all honey samples for extracting pollen DNA. The SOP may be amended as necessary after independent lab validation. The SOP cannot be applied to ultra-filtered honey samples.

3. SUPPLIES, REAGENT AND EQUIPMENT

3.1. Sterile 50 ml conical screw cap centrifuge tubes, copolymer polypropylene (Catalog #1500-1811, USA Scientific)
3.2. Nuclease-Free Water, for Molecular Biology (Catalog #W4502-6x1L, Sigma Aldrich)
3.3. Glass beads (425-600 µm, Catalog #G9268, Sigma)
3.4. DNeasy® Plant Mini Kit (Catalog #69104, Qiagen)
3.5. Proteinase K (Molecular Biology Grade, P8107S, New England BioLabs Inc.)
3.6. QIAquick® PCR Purification Kit (Catalog #28104, Qiagen)
3.7. QIAquick Spin Columns (Catalog #28115, Qiagen)
3.8. QuantiFluor® dsDNA System (Catalog #E2670, Promega)
3.9. Molecular biology grade ethanol (Catalog #BP2818500, Fisher Scientific)
3.10. Seal-Rite 2.0 ml graduated microcentrifuge tube, natural, polypropylene (Catalog #1620-2700, USA Scientific)
3.11. Eppendorf DNA LoBind Tubes, 2.0 ml, PCR clean, colorless (Catalog #4043-1048, USA Scientific)
3.12. Sterile filter microtips (USA Scientific)
    3.10.1. TipOne 0.1-10/20 µl XL natural, graduated filter pipet tips in racks (Catalog #1120-3810)
    3.10.2. TipOne 1-200 µl graduated filter tip refill (Catalog #1120-8710)
    3.10.3. TipOne 1000 µl XL natural, graduated XL filter pipet tips in racks (Catalog #1122-1830)
3.13. Micropipettes (Variable volumes 0.5-10 µl, 10-100 µl and 100-1000 µl)
3.14. Corning® 96-well Black Flat Bottom Polystyrene NBS Microplate (Catalog #3991)
3.15. Weighing scale
3.16. Water bath
3.17. Centrifuge
3.18. Tabletop centrifuge
3.19. Vortex
3.20. Ice box
3.21. 70% ethanol solution
3.22. Tecan plate reader
3.23. Incubator
3.24. Autoclave

4. PROCEDURE

4.1. Important points before starting the procedure

4.1.1. Ensure honey samples to be analyzed are homogeneous. If crystallization of sugar is seen (by eye) in the packaged honey, heat the bottle in an incubator at 50ºC for 30-40 mins to ensure efficient separation of pollen from honey.

4.1.2. Sterilize the glass beads by autoclaving at 121ºC for 30 min.

4.1.3. Set the temperature of the water bath to 65ºC.

4.1.4. Please read the instructions given in the handbook before using the DNeasy® Plant Mini Kit. https://www.qiagen.com/us/resources/resourcedetail?id=95dec8a9-ec37-4457-8884-5dedd8ba9448&lang=en

4.1.5. Add molecular biology grade ethanol to Buffer AW1 and Buffer AW2 concentrates as instructed in the kit.

4.1.6. Please read the instructions given in the handbook before using the QIAquick® PCR Purification Kit. https://www.qiagen.com/us/resources/resourcedetail?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en

4.1.7. Add molecular biology grade ethanol (96–100%) to Buffer PE before use (see bottle label for volume).

4.2. Isolation of pollen from honey

4.2.1. Label a 50 ml conical screw cap centrifuge tube with appropriate code corresponding to the honey sample to be analyzed (Example: H1). Place the tube on a balance and press tare.

4.2.2. Now transfer 15 g of honey into the labeled tube and make up the weight to 50 g using nuclease-free water. Vortex the tube for uniform mixing of the sample.

4.2.3. Incubate the sample at 56ºC for 10-15 min. Vortex the sample in between.

4.2.4. Centrifuge the sample at 4,000 *g* for 30 min at room temperature.

4.2.5. Discard the supernatant without disturbing the pellet. Resuspend the pollen pellet in approximately 1.5 ml nuclease-free water and transfer the sample into a sterile 2 ml microcentrifuge tube. Make sure to scrape the walls of the 50 ml tube using a sterile microtip for complete recovery of the pollen.

4.2.6. Centrifuge the sample transferred in the microcentrifuge tube at 4,000 *g* for 15 min at room temperature.

4.2.7. Carefully remove the supernatant using a sterile microtip without disturbing the pellet. Resuspend the pollen pellet in 100 µl nuclease free water.

4.2.8. Add 7-8 sterile glass beads to the pellet. Vortex the pellet for 2 min on high speed to disrupt the pollens.

4.2.9. Transfer the disrupted the pollen sample to a new microcentrifuge tube without the glass beads. Measure the wet weight of the sample. Proceed to extraction of DNA from pollen or store the sample at -20ºC.

4.3. Extraction of DNA from pollen using DNeasy® Plant Mini Kit

4.3.1. Add 400 µl Buffer AP1 and 25 µl of Proteinase K (20 mg/ml; NEB) to approximately 100 mg of disrupted pollen sample. Vortex and incubate the sample for 10 min at 56 ºC .

4.3.2. Cool at room temperature for 2 min and add 4 µl RNase A and mix gently by inverting the tubes 3-4 times. Incubate the sample for 10 min at 65ºC

4.3.3. Follow rest of the DNA extraction steps (starting from Step 3.) as mentioned in the protocol provided with the kit. Please see PDF of Quick-Start Protocol (version March 2016) attached as supplementary file 1 with the SOP.

4.3.4. At the end of extraction, the crude plant genomic DNA extract is present in 200 µl of Buffer AE.

4.4. Removal of PCR inhibitory components from extracted crude genomic DNA using QIAquick® PCR Purification Kit

4.4.1. Add 1000 µl Buffer PB to 200 µl crude extract of genomic DNA present in Buffer AE. Mix the sample by aspirating 7-8 times using a micropipette.

4.4.2. Follow the rest of the DNA purification steps (starting from Step 2.) as mentioned in the protocol provided with the kit. Please see PDF of Quick-Start Protocol (version July 2018) attached as supplementary file 2 with the SOP.

4.4.3. Finally, elute the genomic DNA from the QIAquick column in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5).

4.4.4. Store the eluted DNA at -20ºC until further use.

5. QUALITY CONTROL OF PURIFIED GENOMIC DNA

Evaluate the concentration of the eluted plant genomic DNA using QuantiFluor® dsDNA System in multiwell plates.

5.1. Follow the DNA quantification steps exactly as mentioned in the protocol provided with the kit (starting from Page 4 to Page 8.). Please see the PDF of the protocol attached as supplementary file 3 to the SOP.

5.2. For optimal results, add at least 5 µl of purified DNA.

5.3. While calculating the concentration of dsDNA, make sure to divide the amount of DNA (ng) obtained from graph by 5 µl to get final value of concentration of DNA in ng/µl (please see page 8 of supplementary text file 3).

5.4. Record the data obtained.

6. APPLICATION OF PURIFIED GENOMIC DNA

The extracted genomic DNA can now readily be used for target amplification of ITS2 by PCR reaction and subsequently for NGS.

7. IMPORTANT NOTES

7.1. Make sure the wet weight of the pollen pellet is between 50-100 mg for efficient extraction of pollen DNA.

7.2. In case of honey samples that have low pollen content, process two or more tubes (each containing 15 g honey sample).
7.3. Avoid frequent vortexing of the extracted genomic DNA on high speed to prevent fragmentation of genomic DNA.
7.4. Always cover the tube and the multiwell plate that contains the Quantifluor dye solution with aluminum foil to prevent bleaching of the fluorescent dye which can affect the sensitivity of the assay.

8. SUPPLEMENTARY FILES

8.1. Quick start protocol for DNeasy® Plant Mini Kit (version March 2016). https://www.qiagen.com/us/resources/resourcedetail?id=6b9bcd96-d7d4-48a1-9838-58dbfb0e57d0&lang=en
8.2. Quick start protocol for QIAquick® PCR Purification Kit (version July 2018). https://www.qiagen.com/us/resources/resourcedetail?id=e0fab087-ea52-4c16-b79f-c224bf760c39&lang=en
8.3. Protocol for QuantiFluor® dsDNA System in multiwell plates (pages 4-8). https://www.promega.com/products/rna-analysis/dna-and-rna-quantitation/quantifluor-dsdna-system/?catNum=E2670#protocols

9. REFERENCES

9.1. Soares, S., Amaral, J. S., Oliveira, M. B. P., & Mafra, I. (2015). Improving DNA isolation from honey for the botanical origin identification. Food Control, 48, 130-136.
9.2. Telfer, E., Graham, N., Stanbra, L., Manley, T., & Wilcox, P. (2013). Extraction of high purity genomic DNA from pine for use in a high-throughput Genotyping Platform. New Zealand Journal of Forestry Science, 43(1), 3.
9.3. Cottenet, G., Blancpain, C., Sonnard, V., & Chuah, P. F. (2013). Development and validation of a multiplex real-time PCR method to simultaneously detect 47 targets for the identification of genetically modified organisms. Analytical and Bioanalytical Chemistry, 405(21), 6831-6844.

SOP Title: AMPLIFICATION OF ITS2 FROM POLLEN FOR NGS			
SOP No. **UH/Honey/02**	**Version No.** **2**	**Effective Date:** **02012020**	**Revision Date:** **07012020**
Written by **Dimple Chavan**	**Reviewed by** **Dr. Katerina Kourentzi**	**Approved by** **Dr. Richard C. Willson**	**Page No.** **Page 1 of 5**

1. PURPOSE

The purpose of this standard operating procedure (SOP) is to describe the procedure for target amplification of ITS2 gene from plant genomic DNA isolated from pollen in honey using Q5 high-fidelity DNA polymerase. The SOP also describes the quality control analysis of purified ITS2 product for downstream application of Next-generation sequencing (NGS).

2. SCOPE

The SOP can be applied to all plant genomic DNA isolated from pollen present in honey samples. The SOP may be amended as necessary after independent lab validation.

3. SUPPLIES, REAGENT AND EQUIPMENT

3.1. Q5® Hot Start High-Fidelity 2X Master Mix (Catalog #M0494S, New England Biolabs Inc.)
3.2. ITS2 primers (Integrated DNA Technologies, Inc.)
  3.2.1. Forward primer (20 bases, 5'- ATG CGA TAC TTG GTG TGA AT -3')
  3.2.2. Reverse primer (21 bases, 5'-GAC GCT TCT CCA GAC TAC AAT-3')
3.3. Nuclease-Free Water, for Molecular Biology (Catalog #W4502-6x1L, Sigma Aldrich)
3.4. QIAquick® PCR Purification Kit (Catalog #28104, Qiagen)
3.5. QIAquick Spin Columns (Catalog #28115, Qiagen)
3.6. QuantiFluor® dsDNA System (Catalog #E2670, Promega)
3.7. Molecular biology grade ethanol (Catalog #BP2818500, Fisher Scientific)
3.8. Agarose Med EEO (Catalog #A1035, US Biological Life sciences)
3.9. Horizontal gel electrophoresis setup assembly (VWR)
3.10. Gel loading buffer Purple (6X) (Catalog #B7024S, New England Biolabs Inc.)
3.11. SYBR™ safe DNA gel stain (Catalog # S33102, ThermoFisher Scientific)
3.12. Gel electrophoresis buffer (1X Tris Acetate EDTA buffer, pH 8.5)
3.13. Agarose gel analyzer system
3.14. Seal-Rite 2.0 ml graduated microcentrifuge tube, natural, polypropylene (Catalog #1620-2700, USA Scientific)
3.15. Eppendorf DNA LoBind Tubes, 2.0 ml, PCR clean, colorless (Catalog #4043-1048, USA Scientific)
3.16. Mx3000P optical strip tubes (Catalog #401428, Agilent Technologies Inc.)
3.17. Mx3000P optical strip caps (Catalog #401425, Agilent Technologies Inc.)
3.18. PCR machine (MJ Mini Thermal Cycler, Bio-Rad)
3.19. PCR hood (Air Clean 600 PCR workstation)
3.20. Sterile filter microtips (USA Scientific)
  3.10.1. TipOne 0.1-10/20 µl XL natural, graduated filter pipet tips in racks (Catalog #1120-3810)
  3.10.2. TipOne 1-200 µl graduated filter tip refill (Catalog #1120-8710)
  3.10.3. TipOne 1000 µl XL natural, graduated XL filter pipet tips in racks (Catalog #1122-1830)
3.21. Micropipettes (Variable volumes 0.5-10 µl, 10-100 µl and 100-1000 µl)
3.22. Corning® 96-well Black Flat Bottom Polystyrene NBS Microplate (Catalog #3991)
3.23. Tabletop centrifuge

3.24. Vortex
3.25. Ice box
3.26. 70% ethanol solution
3.27. DNAZap solution (Catalog #AM9890, Invitrogen)
3.28. Nanodrop
3.29. Tecan plate reader
3.30. Autoclave

4. PROCEDURE

4.1. Important points before starting the procedure

4.1.1. Dilute the 100 µM main stock solutions of forward primer and reverse primer to a working stock solution having concentration of 10 µM each using nuclease free water. Store both the stock solutions of primers at -20ºC.
4.1.2. Sterilize the microcentrifuge tubes by autoclaving at 121ºC for 30 min.
4.1.3. Please read the instructions given in the handbook before using the QIAquick® PCR Purification Kit. https://www.qiagen.com/us/resources/resourcedetail?id=95f10677-aa29-453d-a222-0e19f01ebe17&lang=en
4.1.4. Add molecular biology grade ethanol (96–100%) to Buffer PE before use (see bottle label for volume).

4.2. Setting up the PCR reaction

4.2.1. Before setting up the PCR reaction please follow routine cleanup process of PCR hood. Sterilize the working area of PCR hood by UV sterilization for 15 min.
4.2.2. Clean the micropipettes and working space with 70% ethanol solution.
4.2.3. Label the PCR tubes with the appropriate codes. Allow the PCR reaction components (PCR master mix, primers and DNA template) to thaw completely before mixing them together.
4.2.4. Mix the components as given in the table below.

Component	50 µl Reaction	Final concentration
Q5 High-Fidelity 2X Master Mix	25 µl	1X
10 µM Forward Primer (ITS2)	2.5 µl	0.5 µM
10 µM Reverse Primer (ITS2)	2.5 µl	0.5 µM
Template DNA	2 µl	
Nuclease-Free Water	18 µl	

Notes: Gently mix the reaction. Collect all liquid to the bottom of the tube by a quick spin if necessary.

4.2.5. For every honey sample to be analyzed, prepare two PCR tubes (each tube containing 2 µl DNA template isolated from that honey; 10-50 ng DNA).
4.2.6. Prepare a no-template control reaction tube by mixing all components as mentioned in the above table except DNA template. Add 2 µl nuclease free water instead of 2 µl DNA template.
4.2.7. Transfer PCR tubes to a PCR machine and begin thermocycling.

Thermocycling conditions for ITS2 amplification

Step	Temperature	Time
Initial Denaturation	98°C	30 seconds
40 Cycles	98°C	10 seconds
	62°C	30 seconds
	72°C	30 seconds
Final Extension	72°C	2 minutes
Hold	4°C	

4.2.8. Alternatively, for samples that fail to amplify above conditions. Modified program can be applied for ITS2 amplification.

Step	Temperature	Time
Initial Denaturation	98°C	30 seconds
40 Cycles	98°C	10 seconds
	62°C	30 seconds
	72°C	1 minute
Final Extension	72°C	5 minutes
Hold	4°C	

4.3. Purification of ITS2 PCR product using QIAquick® PCR Purification Kit

4.3.1. Pool the two PCR tubes (containing the amplified ITS2 product) corresponding to the same honey sample together by transferring the contents in a single sterile 2 ml DNA LoBind tube.

4.3.2. For every honey sample which is pooled you will have 100 µl amplified ITS2 product.

4.3.3. Add 500 µl Buffer PB to 100 µl amplified ITS2 product. Mix the sample by aspirating 7-8 times using a micropipette.

4.3.4. Follow the rest of the DNA purification steps as mentioned in the protocol provided with the kit. Please see PDF of Quick-Start Protocol (version July 2018) attached as supplementary file 2 with the SOP.

4.3.5. Finally, elute the purified ITS2 product from the QIAquick column in 50 µl Buffer EB (10 mM Tris·Cl, pH 8.5) in a sterile 2 ml DNA LoBind Tube.

4.3.6. Store the purified ITS2 product at -20ºC until further use.

5. QUALITY CONTROL OF PURIFIED ITS2 PRODUCT

5.1. Agarose gel electrophoresis of amplified ITS2 product

5.1.1. Prepare 1.5% agarose gel in 1X TAE buffer (pH 8.5).

5.1.2. Mix 5 µl purified ITS2 product with 1 µl of 6X gel loading buffer.

5.1.3. Place the solidified agarose gel in an electrophoresis tank containing 1X TAE buffer (pH 8.5). Make sure gel is completely immersed in the buffer.

5.1.4. Tabulate in which order the samples will be loaded in the gel to avoid any confusion later.

5.1.5. Load 6 µl of ITS2 product mixed with gel loading buffer into the gel. Load 9 µl of the Hi-Lo DNA ladder in the first and last well so analyzing the samples is easy. Avoid introducing any air bubbles while loading the samples.

5.1.6. Run the gel electrophoresis at 96-100 V for 45 min to 1 h until the tracking dye reaches 3/4th of the gel height.

5.1.7. Turn off the power and transfer the gel in a plastic box containing 100 ml 1X TAE and 10 µl of SYBR™ Safe DNA Gel Stain for staining the bands.

5.1.8. Protect the staining container from light by covering it with aluminum foil or placing it in the dark. Agitate the gel gently at room temperature for 30 min.

5.1.9. Observe the stained gel using gel analyzer. You should see multiple bands of ITS2 ranging from 150-500 bp as honey can be a blend from different plants. Save the results obtained.

5.2. **DNA quantification using QuantiFluor® dsDNA System**

5.2.1. Evaluate the concentration of amplified ITS2 product (purified) using QuantiFluor® dsDNA System in multiwell plates.

5.2.2. Follow the DNA quantification steps exactly as mentioned in the protocol provided with the kit (from Page 4 to Page 8.). Please see PDF of protocol attached as supplementary file 3 with the SOP.

5.2.3. For optimal results, add at least 1-2 µl of amplified ITS2 product.

5.2.4. While calculating the concentration of dsDNA, make sure to divide the amount of DNA (ng) obtained from the graph by the volume of DNA (µl) that was added to the well to get final value of concentration of DNA in ng/µl (please see page 8 of supplementary text file 3).

5.2.5. Record the data obtained.

5.3. Using Nanodrop

5.3.1. Evaluate the quality of the eluted ITS2 product using Nanodrop A260 value, and 260/280 and 260/230 ratios.

5.3.2. Record all the values for honey samples.

6. NEXT GENERATION SEQUENCING OF PURIFIED ITS2 PRODUCT

6.1. Prepare the samples for NGS as per criteria mentioned on the sample submission guidelines by Genewiz.
https://www.genewiz.com/Public/Services/Next-Generation-Sequencing/Amplicon-Sequencing-Services/Amplicon-EZ

Next Generation Sequencing: Amplicon-EZ Sample Type: Purified PCR Products Minimum Amount: 500 ng Concentration: Normalized to 20 ng/µl Purity (A260/280): 1.8-2.0 Buffer: Water, EB, or low TE (<0.1 mM EDTA)

6.2. If the concentration of purified ITS2 product obtained is greater than 20 ng/µl then normalize it to the required concentration (20 ng/µl) using Buffer EB.
6.3. Make sure you send sufficient amount of ITS2 product for sequencing, usually between 500-600 ng.
6.4. Label all samples with appropriate code and make a note of it.
6.5. If you are shipping the samples after more than 2 days, make sure you store all DNA products at -20°C.

7. IMPORTANT NOTES

7.1. Make sure you have a separate set of micropipettes in the PCR hood. Micropipettes from this hood must not be taken out.
7.2. Clean the PCR hood at the end of work with DNAZap to prevent carryover of DNA contamination for next run.
7.3. Alternatively, you can use 1X Sodium borate buffer (pH 8.5) for gels and running buffer for electrophoresis for better resolution of smaller size DNA bands (between 50-300 bp).
7.4. Always cover the tube and the multiwell plate that contains the Quantifluor dye solution with aluminum foil to prevent bleaching of the fluorescent dye, which can affect the sensitivity of the assay.

8. SUPPLEMENTARY FILES

8.1. Quick start protocol for QIAquick® PCR Purification Kit (version July 2018). https://www.qiagen.com/us/resources/resourcedetail?id=e0fab087-ea52-4c16-b79f-c224bf760c39&lang=en
8.2. Protocol for QuantiFluor® dsDNA System in multiwell plates (pages 4-8). https://www.promega.com/products/rna-analysis/dna-and-rna-quantitation/quantifluor-dsdna-system/?catNum=E2670#protocols

9. REFERENCES

9.1. Chen, S., Yao, H., Han, J., Liu, C., Song, J., Shi, L., ... & Luo, K. (2010). Validation of the ITS2 region as a novel DNA barcode for identifying medicinal plant species. *PloS one*, *5*(1).

Acknowledgment: This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 17STBTI00001-03-00, formerly 2015-ST-061-BSH001.


U.S. DEPARTMENT OF SECURITY