

SUPPLEMENTARY MATERIAL 1

MySQL query that generated the protochecklist of N.A. nonlichenized Fungi:

```
select DISTINCT t2.SciName, t2.Author as scientificNameAuthorship, t3.SciName as
ParentName
from omoccurr_na_spec as o left join taxa as t on o.sciname = t.SciName
left join taxstatus as ts on t.TID = ts.tid
left join taxa as t2 on ts.tidaccepted = t2.TID
left join taxaenumtree as te on ts.tidaccepted = te.tid
left join taxa as t3 on te.parenttid = t3.TID
where ts.taxauthid = 2 and te.taxauthid = 2 and
(t3.SciName
in("Ascomycota","Basidiomycota","Blastocladiomycota","Chytridiomycota","Fungi
Incertae Sedis","Glomeromycota","Zygomycota"))
order by t2.SciName;
```

MySQL query that generated the list of N.A. nonlichenized fungal type specimens:

```
select o.occid, o.collid, o.catalogNumber, IFNULL(t2.SciName,o.sciname) as sciname,
        ts.tidaccepted as tidinterpreted,
IFNULL(t2.Author,o.scientificNameAuthorship) as scientificNameAuthorship,
```

```

        o.typeStatus, o.occurrenceRemarks, o.country
from omoccurrences as o left join taxa as t on o.sciname = t.SciName
left join taxstatus as ts on t.TID = ts.tid
left join taxa as t2 on ts.tidaccepted = t2.TID
where ts.taxauthid = 2 and o.typeStatus is not null and
(o.country = "United States" or o.country = "Canada" or o.country = "Mexico")
order by sciname;

```

Script for removing genera in the N.A. nonlichenized Fungi lists:

```

DROP TABLE IF EXISTS List;

```

```

.mode csv
-- .import tmp/NA_taxa_list_IF_editted.csv List
.import tmp/NA_taxa_list_MB_editted.csv List
.mode ascii

```

```

DROP TABLE IF EXISTS Audit;

```

```

CREATE TABLE Audit AS

```

```

SELECT SciName, CASE WHEN length(SciName) >= 1

```

```

THEN

```

```

    (length(SciName) - length(replace(SciName, ' ', ''))) + 1

```

ELSE

(length(SciName) - length(replace(SciName, ' ', '')))

END as NumOfWords FROM List;

DROP TABLE IF EXISTS Scinames;

CREATE TABLE Scinames AS

SELECT DISTINCT(SciName) FROM Audit

WHERE NumOfWords > 1;

DROP TABLE IF EXISTS NewList;

CREATE TABLE NewList AS

SELECT *

FROM List

WHERE SciName IN (SELECT SciName FROM Scinames);

.mode list

SELECT count(*) AS TotalOriginalRecords FROM List;

SELECT count(*) AS TotalFilteredRecords FROM NewList;

.headers on

.mode csv

-- .output tmp/NA_taxa_list_IF_editted_NEW_.csv

.output tmp/NA_taxa_list_MB_editted_NEW_.csv

```
SELECT * FROM NewList;
```

Script for removing duplicate names with preference for holotypes in the list of N.A. type specimens:

```
DROP TABLE IF EXISTS List;
```

```
.mode csv
```

```
.import tmp/type_na_MB_query_editted_FINAL.csv List
```

```
.mode ascii
```

```
-- Extract all entries where typeStatus contains holotype
```

```
DROP TABLE IF EXISTS ContainsHolotype;
```

```
CREATE TABLE ContainsHolotype AS
```

```
SELECT * FROM List WHERE typeStatus LIKE '%holotype%';
```

```
-- Extract all entries where typeStatus contains holotype
```

```
DROP TABLE IF EXISTS DoesNotContainsHolotype;
```

```
CREATE TABLE DoesNotContainsHolotype AS
```

```
SELECT * FROM List WHERE sciname NOT IN (SELECT distinct(sciname) FROM  
ContainsHolotype);
```

-- Combine results

DROP TABLE IF EXISTS NewList;

CREATE TABLE NewList AS

SELECT * FROM (

SELECT * FROM ContainsHolotype GROUP BY sciname

UNION All

SELECT * FROM DoesNotContainsHolotype GROUP BY sciname

) ORDER BY sciname ASC;

.mode list

-- TEST counts

select count(distinct(sciname)) AS Original_UniqueSpecies FROM List;

select count(distinct(sciname)) AS Filtered_UniqueSpecies FROM NewList;

select count(distinct(sciname)) AS Holotype_UniqueSpecies FROM ContainsHolotype;

select count(distinct(sciname)) AS OtherStatus_UniqueSpecies FROM

DoesNotContainsHolotype;

.headers on

.mode csv

.output tmp/type_na_MB_query_editted_FINAL__NEW_.csv

SELECT * FROM NewList;