

# Retrieving Accepted Names from Kew's Plants of the World and Assigning an IPNI Identification Number

Keisey Brock  
2/18/2020

## Part 1: Getting POW Accepted Names and IPNI IDS for each Taxon

### Step 1: Set Up R

<pre>require(knitr)</pre>
<pre>## Loading required package: knitr</pre>
<pre>knitr::opts_knit\$set(root.dir = "C:/Users/ksaley/Documents/work/hapi/") #set the working directory</pre>
<pre>## Install needed packages if (!require("pacman")){   install.packages("pacman")   library(pacman) }</pre>
<pre>## Loading required package: pacman</pre>
<pre>p_load("taxize", "dplyr", "reshape2", "data.table",         "rctdwr", "sqldf", "knitr", "kableExtra")</pre>

### Step 2: Read in a List of Species Names

<pre>## Read in a csv file with your species on it. applist &lt;- read.csv("sample_app_list.csv", header=T, asp=",", stringsAsFactors=F) head(applist, 10)</pre>
<pre>##           Taxa ## 1 Bignonia tweediana ## 2 Dolichandra unguis-cati ## 3 Acacia farnesiana ## 4 Acacia koa ## 5 Aristolochia philippinensis ## 6 Breytia vitis-idaea ## 7 Citrus "Shasta Gold" ## 8 Citrus unknownum ## 9 Colysis amplia ## 10 Chamaecrista nictitans subsp. patellaria ## 11 Chamaecrista nictitans subsp. patellaria var. glabrata ## 12 Citrus aurantium ## 13 Asystasia gangetica subsp. micrantha ## 14 Eleanoreopsis angustifolius ## 15 Fouquieria columnaris (Kellogg) Kellogg ex Curran ## 16 Heliconia hirsuta ## 17 Hibiscus columariatus ## 18 Jasminum rex ## 19 Melia azadirach ## 20 Odontonia cuspidatum ## 21 Phragmanthera capitata ## 22 Pritchardia liliolum dulcis ## 23 Pylurus banksii ## 24 Tavernia quinquevittata ## 25 Vigna sandwicensis</pre>
<ul style="list-style-type: none"><li>In this sample data set of 25 taxa, I have included misspellings (e.g. Vigna sandwicensis), subspecies (Chamaecrista nictitans subsp. patellaria), varieties (Chamaecrista nictitans subsp. patellaria var. glabrata), cultivars (Citrus "Shasta Gold"), and things that likely won't get a database hit (Citrus unknownum). There's also an example of bad capitalization and punctuation (asystasia gangetica subsp micrantha) and an instance where the authority is included (Fouquieria columnaris (Kellogg) Kellogg ex Curran).</li></ul>

### Step 3: Make sure each name is a valid synonym (even if it's not the "accepted" name that you ultimately want)

- This step will also take care of misspellings and capitalizations of genus versus specific epithets

Checking for taxonomic reference databases; we're looking for The International Plant Names Index

<pre>gwr_datasources()</pre>
<pre>## # A tibble: 100 x 12 ##   created_at data_hash data_url description      id log_url ##   &lt;date&gt;    &lt;dbl&gt; &lt;dbl&gt;    &lt;chr&gt;      &lt;chr&gt;      &lt;chr&gt; ## 1 2012-07-01 NA      NA      "(*This z- 1 NA ## 2 2012-07-01 NA      NA      "The tree a- 3 NA ## 3 2012-02-01 NA      NA      "The white ~ 3 NA ## 4 2012-02-01 NA      NA      "The Nation- 4 NA ## 5 2012-02-01 NA      NA      "Classifica- 3 NA ## 6 2012-02-01 NA      NA      "GRIN taxon- 6 NA ## 7 2012-02-01 NA      NA      "This class- 3 NA ## 8 2012-02-01 NA      NA      "The inter- 8 NA ## 9 2012-02-01 NA      NA      "An authori- 9 NA ## 10 2012-02-01 NA      NA      "An entity ~ 10 NA ## # ... with 90 more rows, and 6 more variables: name_strings_count &lt;int&gt;, ## #   refresh_period_days &lt;int&gt;, title &lt;chr&gt;, unique_names_count &lt;int&gt;, ## #   updated_at &lt;chr&gt;, web_site_url &lt;chr&gt;</pre>
<pre># IPNI is # 167</pre>

Now we run the Global Names Resolver, asking it only return the nearest matching IPNI name

<pre>gwr.long &lt;- applist\$Taxa %&gt;% gwr_resolve(data_source_id = 167, best_match_only=T, with_canonical_ranks=T)</pre>
<pre>head(gwr.long)</pre>
<pre>## # A tibble: 6 x 5 ##   user_supplied_name submitted_name data_source title score matched_name2 ##   &lt;chr&gt;                &lt;chr&gt;      &lt;chr&gt;      &lt;chr&gt;      &lt;dbl&gt;      &lt;chr&gt; ## 1 Bignonia tweediana Bignonia tweedi- The Internationa- 0.73 Bignonia twee- ## 2 Dolichandra ungu- Dolichandra ungu- The Internationa- 0.73 Dolichandra u- ## 3 Acacia farnesiana Acacia farnesia- The Internationa- 0.988 Acacia farnes- ## 4 Acacia koa          Acacia koa      The Internationa- 0.988 Acacia koa ## 5 Aristolochia phil- Aristolochia ph- The Internationa- 0.988 Aristolochia ~ ## 6 Breytia vitis-ida- Breytia vitis-i- The Internationa- 0.988 Breytia vitis-</pre>
<ul style="list-style-type: none"><li>You can see that it returns a column called "matched_name2" with the viable synonym.</li><li>Because the Global Names Resolver can be time consuming for long species lists, it's probably wise to save the output as a csv</li></ul>
<pre>## Writing to csv file write.csv(gwr.long,           "samplegwr.csv",           row.names = F, quote = F)</pre>
<pre>## Reading it in again, in case you had to restart R for some reason gwr.long &lt;- read.csv("samplegwr.csv",                     header=T, stringsAsFactors = F)</pre>

### Step 4: Return an ID Number that Matches the Accepted Name (According to Plants of the World)

- This step will prompt you if the database is unsure about which name to return. You have to specify the row number of taxa you want to select as it runs through them.

<pre>## Fetching IPNI ids from POW pow.output &lt;- get_pow(gwr.long\$matched_name2, db = "pow", accepted = TRUE, ask = TRUE)</pre>
<pre>head(pow.output)</pre>
<pre>## [1] "urn:lsid:ipni.org:names:77100466-1" ## [2] "urn:lsid:ipni.org:names:77100466-1" ## [3] NA ## [4] "urn:lsid:ipni.org:names:47069-1" ## [5] "urn:lsid:ipni.org:names:93225-1" ## [6] "urn:lsid:ipni.org:names:340112-1"</pre>

-You can see that and IPNI id has been returned in the "ids" column. -Because the get\_pow() function can be time consuming for long species lists, it's probably wise to save the output as a csv.

<pre>## Writing to csv file write.csv(as.data.frame(pow.output),           "sample_pow_output.csv",           row.names = F, quote = F)</pre>
<pre>## Reading it in again, in case you had to restart R for some reason pow.output &lt;- read.csv("sample_pow_output.csv",                       header=T, stringsAsFactors = F)</pre>

-Now to combine the IPNI IDs back to the original names

<pre>inputname_ID &lt;- as.data.frame(cbind(gwr.long\$user_supplied_name, gwr.long\$matched_name2, pow.output\$ids)) head(inputname_ID)</pre>
<pre>colnames(inputname_ID) = c("user_supplied_name", "matched_name2", "IPNI_ID") head(inputname_ID)</pre>
<pre>##           user_supplied_name matched_name2 ## 1 Bignonia tweediana      Bignonia tweediana ## 2 Dolichandra unguis-cati Dolichandra unguis-cati ## 3 Acacia farnesiana      Acacia farnesiana ## 4 Acacia koa              Acacia koa ## 5 Aristolochia philippinensis Aristolochia philippinensis ## 6 Breytia vitis-idaea      Breytia vitis-idaea ## 7 IPNI_ID ## 1 urn:lsid:ipni.org:names:77100466-1 ## 2 urn:lsid:ipni.org:names:77100466-1 ## 3 urn:lsid:ipni.org:names:47069-1 ## 4 urn:lsid:ipni.org:names:47069-1 ## 5 urn:lsid:ipni.org:names:93225-1 ## 6 urn:lsid:ipni.org:names:93225-1 ## 7 urn:lsid:ipni.org:names:340112-1</pre>

### Step 5: Separate the troublemakers

<pre>## Those names have returned an IPNI ID inputname_ID.\$ndrop &lt;- subset(inputname_ID, (is.na(inputname_ID\$IPNI_ID))) rown(inputname_ID.\$ndrop)</pre>
<pre>## [1] 18</pre>
<pre>## Those names have NOT returned an IPNI ID and need special human attention inputname_ID.\$family &lt;- subset(inputname_ID, (is.na(inputname_ID\$IPNI_ID))) rown(inputname_ID.\$family)</pre>
<pre>## [1] 7</pre>
<pre>## Writing the trouble makers to csv file (deal with them later) write.csv(inputname_ID.\$family,           "TM_WebdataManualAttention.csv",           row.names = F, quote = F)</pre>

### Step 6: Use the IPNI ID to return data (name, rank, family, genus, native range, etc)

First, we'll get taxon name

<pre>## Writing a function that will return the taxon name associated with the IPNI ID according to POW get_taxonname &lt;- function(ipnames){   temp &lt;- c()   temp &lt;- pow_lookup(ipnames)   ifelse(is.null(temp\$meta\$name)) == "TRUE", "ERROR", temp\$meta\$name }</pre>
<pre>## Applying the function to your list POW.taxonname &lt;- apply(inputname_ID.\$ndrop\$IPNI_ID[, ], Get_taxonname)</pre>

Second, we'll get family

<pre>## Writing a function that will return the family name associated with the IPNI ID according to POW get_family &lt;- function(ipnames){   temp &lt;- c()   temp &lt;- pow_lookup(ipnames)   ifelse(is.null(temp\$meta\$family)) == "TRUE", "ERROR", temp\$meta\$family }</pre>
<pre>## Applying the function to your list POW.family &lt;- apply(inputname_ID.\$ndrop\$IPNI_ID[, ], Get_family)</pre>

Third, we'll get genus

<pre>## Writing a function that will return the genus name associated with the IPNI ID according to POW get_genus &lt;- function(ipnames){   temp &lt;- c()   temp &lt;- pow_lookup(ipnames)   ifelse(is.null(temp\$meta\$genus)) == "TRUE", "ERROR", temp\$meta\$genus }</pre>
<pre>## Applying the function to your list POW.genus &lt;- apply(inputname_ID.\$ndrop\$IPNI_ID[, ], Get_genus)</pre>

Fourth, we'll get rank

<pre>## Writing a function that will return the taxon rank associated with the IPNI ID according to POW get_taxonrank &lt;- function(ipnames){   temp &lt;- c()   temp &lt;- pow_lookup(ipnames)   ifelse(is.null(temp\$meta\$rank)) == "TRUE", "ERROR", temp\$meta\$rank }</pre>
<pre>## Applying the function to your list # this will return a bunch of red text warnings, but disregard POW.taxonrank &lt;- apply(inputname_ID.\$ndrop\$IPNI_ID[, ], Get_taxonrank)</pre>

Fifth, we'll get author

<pre>## Writing a function that will return the taxon rank associated with the IPNI ID according to POW get_author &lt;- function(ipnames){   temp &lt;- c()   temp &lt;- pow_lookup(ipnames)   ifelse(is.null(temp\$meta\$author)) == "TRUE", "ERROR", temp\$meta\$author }</pre>
<pre>## Applying the function to your list # this will return a bunch of red text warnings, but disregard POW.author &lt;- apply(inputname_ID.\$ndrop\$IPNI_ID[, ], Get_author)</pre>

### Step 7: combining the original names, IPNI IDs and other data into a single spreadsheet

<pre>## Writing the data attached to the IPNI ID in one place POWinput &lt;- cbind(POW.family, POW.genus, POW.taxonname, POW.author, POW.taxonrank) POWinput &lt;- as.data.frame(POWinput) head(POWinput)</pre>
<pre>##           POW.family POW.genus POW.taxonname ## 1 Bignoniaceae Dolichandra Dolichandra unguis-cati ## 2 Bignoniaceae Dolichandra Dolichandra unguis-cati ## 3 Fabaceae Acacia Acacia koa ## 4 Aristolochiaceae Aristolochia Aristolochia philippinensis ## 5 Phyllanthaceae Breytia Breytia vitis-idaea ## 6 Polyodiaceae Leptochilus Leptochilus amplius ## 7 POW.author POW.taxonrank ## 1 (L.) L.G.Lohmann SPECIES ## 2 (L.) L.G.Lohmann SPECIES ## 3 A.Gray SPECIES ## 4 Warb. SPECIES ## 5 (Burm.f.) C.B.C.Fisch. SPECIES ## 6 (F.Muell.) Nees. SPECIES</pre>
<pre>POWacc_IPNIid &lt;- cbind(inputname_ID.\$ndrop, POWinput) head(POWacc_IPNIid)</pre>
<pre>##           user_supplied_name matched_name2 ## 1 Bignonia tweediana Bignonia tweediana ## 2 Dolichandra unguis-cati Dolichandra unguis-cati ## 3 Acacia koa          Acacia koa ## 4 Aristolochia philippinensis Aristolochia philippinensis ## 5 Breytia vitis-idaea Breytia vitis-idaea ## 6 Colysis amplia      Colysis amplia ## 7 IPNI_ID ## 1 urn:lsid:ipni.org:names:77100466-1 Bignoniaceae Dolichandra ## 2 urn:lsid:ipni.org:names:77100466-1 Bignoniaceae Dolichandra ## 3 urn:lsid:ipni.org:names:47069-1 Fabaceae Acacia ## 4 urn:lsid:ipni.org:names:93225-1 Aristolochiaceae Aristolochia ## 5 urn:lsid:ipni.org:names:340112-1 Phyllanthaceae Breytia ## 6 urn:lsid:ipni.org:names:17568150-1 Polyodiaceae Leptochilus ## 7 POW.taxonname POW.author POW.taxonrank ## 1 Dolichandra unguis-cati (L.) L.G.Lohmann SPECIES ## 2 Dolichandra unguis-cati (L.) L.G.Lohmann SPECIES ## 3 Acacia koa A.Gray SPECIES ## 4 Aristolochia philippinensis Warb. SPECIES ## 5 Breytia vitis-idaea (Burm.f.) C.B.C.Fisch. SPECIES ## 6 Leptochilus amplius (F.Muell.) Nees. SPECIES</pre>

### Step 8: labelling the names that returned a duplicatedIPNI ID and POW taxonname

- This is especially useful if you have many synonymous names in your user supplied names list

<pre>## POWacc_IPNIid &lt;- data.table( POWacc_IPNIid ) POWacc_IPNIid &lt;- POWacc_IPNIid[, duplicateCount := 1:N, by = "IPNI_ID"] head(POWacc_IPNIid)</pre>
<pre>##           user_supplied_name matched_name2 ## 1: Bignonia tweediana Bignonia tweediana ## 2: Dolichandra unguis-cati Dolichandra unguis-cati ## 3: Acacia koa          Acacia koa ## 4: Aristolochia philippinensis Aristolochia philippinensis ## 5: Breytia vitis-idaea Breytia vitis-idaea ## 6: Colysis amplia      Colysis amplia ## 7: IPNI_ID ## 1: urn:lsid:ipni.org:names:77100466-1 Bignoniaceae Dolichandra ## 2: urn:lsid:ipni.org:names:77100466-1 Bignoniaceae Dolichandra ## 3: urn:lsid:ipni.org:names:47069-1 Fabaceae Acacia ## 4: urn:lsid:ipni.org:names:93225-1 Aristolochiaceae Aristolochia ## 5: urn:lsid:ipni.org:names:340112-1 Phyllanthaceae Breytia ## 6: urn:lsid:ipni.org:names:17568150-1 Polyodiaceae Leptochilus ## 7: POW.taxonname POW.author POW.taxonrank ## 1: Dolichandra unguis-cati (L.) L.G.Lohmann SPECIES ## 2: Dolichandra unguis-cati (L.) L.G.Lohmann SPECIES ## 3: Acacia koa A.Gray SPECIES ## 4: Aristolochia philippinensis Warb. SPECIES ## 5: Breytia vitis-idaea (Burm.f.) C.B.C.Fisch. SPECIES ## 6: Leptochilus amplius (F.Muell.) Nees. SPECIES ## 7: duplicateCount ## 1: 1 ## 2: 2 ## 3: 1 ## 4: 1 ## 5: 1 ## 6: 1</pre>

### Done! ... wrting to csv file

<pre>## Writing to csv file write.csv(POWacc_IPNIid,           "POWacc_IPNIid.csv",           row.names = F)</pre>
--

## Part 2: Getting IPNI IDs for Families and Genera in your species list

### Step 1: Compile a list of both families and genera using your previous output

<pre>## families &lt;- as.character(POWacc_IPNIid\$POW.family) genera &lt;- as.character(POWacc_IPNIid\$POW.genus) head(families, 10)</pre>
<pre>## [1] "Bignoniaceae" "Bignoniaceae" "Fabaceae" ## [4] "Aristolochiaceae" "Phyllanthaceae" "Polyodiaceae" ## [7] "Fabaceae" "Anacardiaceae" "Acanthaceae" ## [10] "Eleanoreopsis" "Fouquieriaceae" " ## [13] "Celastraceae" "Acanthaceae" "Loranthaceae" ## [16] "Proteaceae" "Rubiaceae" "Fabaceae"</pre>
<pre>## Remove duplicates fam.dup.removed &lt;- unique(families) gen.dup.removed &lt;- unique(genera) fam.gen &lt;- c(fam.dup.removed, gen.dup.removed) length(fam.dup.removed)</pre>
<pre>## [1] 14</pre>
<pre>length(gen.dup.removed)</pre>
<pre>## [1] 17</pre>

### Step 2: Fetch IPNI IDs and associated data for families

```
fam.IDs <- get_pow(fam.dup.removed, db = "pow", rank_filter = "family", accepted = TRUE, ask = TRUE)
```

### Step 3: Fetch IPNI IDs and associated data for genera

```
gen.IDs <- get_pow(gen.dup.removed, db = "pow", rank_filter = "genus", accepted = TRUE, ask = TRUE)
```

<pre>fam.IDs.df &lt;- as.data.frame(fam.IDs) fam.IDs.df &lt;- cbind(fam.dup.removed, fam.IDs.df\$ids) colnames(fam.IDs.df) &lt;- c("names", "IPNI_ID") gen.IDs.df &lt;- as.data.frame(gen.IDs) gen.IDs.df &lt;- cbind(gen.dup.removed, gen.IDs.df\$ids) colnames(gen.IDs.df) &lt;- c("names", "IPNI_ID") head(gen.IDs.df)</pre>
<pre>##           names IPNI_ID ## 1 "Dolichandra" "urn:lsid:ipni.org:names:4063-1" ## 2 "Acacia" NA ## 3 "Aristolochia" "urn:lsid:ipni.org:names:330834-2" ## 4 "Breytia" "urn:lsid:ipni.org:names:37312-2" ## 5 "Leptochilus" "urn:lsid:ipni.org:names:1732410-1" ## 6 "Chamaecrista" "urn:lsid:ipni.org:names:323939-2"</pre>
<pre>## Combine genera and families into a single dataframe fam.gen.IDs.df &lt;- rbind(fam.IDs.df, gen.IDs.df) fam.gen.IDs.df &lt;- as.data.frame(fam.gen.IDs.df) rown(fam.gen.IDs.df)</pre>
<pre>## [1] 31</pre>

### Step 4: Separate the troublemakers

<pre>## Those names have returned an IPNI ID fam.gen.ID.\$ndrop &lt;- subset(fam.gen.IDs.df, (is.na(fam.gen.IDs.df\$IPNI_ID))) rown(fam.gen.ID.\$ndrop)</pre>
<pre>## [1] 20</pre>
<pre>## Those names have NOT returned an IPNI ID and need special human attention fam.gen.IDs.\$family &lt;- subset(fam.gen.IDs.df, (is.na(fam.gen.IDs.df\$IPNI_ID))) rown(fam.gen.IDs.\$family)</pre>
<pre>## [1] 11</pre>
<pre>## Writing the trouble makers to csv file (deal with them later) write.csv(fam.gen.IDs.\$family,           "TM_Fam_Genera_WebdataManualAttention.csv",           row.names = F, quote = F)</pre>

### Step 5: Fetch data from POW database using IPNI ID

<pre>## POW.fam.gen.name &lt;- apply(fam.gen.ID.\$ndrop\$IPNI_ID[, ], Get_taxonname) POW.fam.gen.author &lt;- apply(fam.gen.ID.\$ndrop\$IPNI_ID[, ], Get_author) POW.fam.gen.rank &lt;- apply(fam.gen.ID.\$ndrop\$IPNI_ID[, ], Get_taxonrank) POW.fam.gen.family &lt;- apply(fam.gen.ID.\$ndrop\$IPNI_ID[, ], Get_family)</pre>
<pre>## Writing the data attached in one place POWinput.fam.gen &lt;- cbind(POW.fam.gen.family, POW.fam.gen.name, POW.fam.gen.author, POW.fam.gen.rank) POWinput.fam.gen &lt;- as.data.frame(POWinput.fam.gen) head(POWinput.fam.gen)</pre>
<pre>##           POW.fam.gen.family POW.fam.gen.name POW.fam.gen.author POW.fam.gen.rank ## 1 Bignoniaceae Bignoniaceae Juss. FAMILY ## 2 Aristolochiaceae Aristolochiaceae J.Presl &amp; C.Presl FAMILY ## 3 Polyodiaceae Polyodiaceae J.Presl &amp; C.Presl FAMILY ## 4 Anacardiaceae Anacardiaceae J.St.-Hil. FAMILY ## 5 Acanthaceae Acanthaceae Juss. FAMILY ## 6 Celastraceae Celastraceae Hoffmanns. &amp; Link FAMILY</pre>
<pre>POWacc_IPNIid.fam.gen &lt;- cbind(fam.gen.ID.\$ndrop, POWinput.fam.gen) head(POWacc_IPNIid.fam.gen)</pre>
<pre>##           names IPNI_ID POW.fam.gen.family ## 1 Bignoniaceae urn:lsid:ipni.org:names:30000204-2 Bignoniaceae ## 2 Aristolochiaceae urn:lsid:ipni.org:names:30000309-2 Aristolochiaceae ## 3 Polyodiaceae urn:lsid:ipni.org:names:30000471-2 Polyodiaceae ## 4 Anacardiaceae urn:lsid:ipni.org:names:30000359-2 Anacardiaceae ## 5 Acanthaceae urn:lsid:ipni.org:names:30000416-2 Acanthaceae ## 6 Celastraceae urn:lsid:ipni.org:names:30001336-2 Celastraceae ## 7 POW.fam.gen.name POW.fam.gen.author POW.fam.gen.rank ## 1 Bignoniaceae Juss. FAMILY ## 2 Aristolochiaceae Juss. FAMILY ## 3 Polyodiaceae J.Presl &amp; C.Presl FAMILY ## 4 Anacardiaceae J.St.-Hil. FAMILY ## 5 Acanthaceae Juss. FAMILY ## 6 Celastraceae Hoffmanns. &amp; Link FAMILY</pre>

### Done! ... wrting to csv file

<pre>## Writing to csv file write.csv(POWacc_IPNIid.fam.gen,           "POWacc_IPNIid.fam.gen.csv",           row.names = F)</pre>
--