00000100000
00001010000
00010001000
00100000100
01000000010
10000000001
Software notes

Ecography 38: 426–430, 2015
doi: 10.1111/ecog.01055

Subject Editor: Brody Sandel. Editor-in-Chief: Miguel Araújo. Accepted 11 December 2014

NCBIminer: sequences harvest from Genbank

Xiaoting Xu, Dimitar Dimitrov, Carsten Rahbek and Zhiheng Wang

X. Xu and Z. Wang (zhiheng.wang@pku.edu.cn), Dept of Ecology and Key Laboratory for Earth Surface Processes of the Ministry of Education, College of Urban and Environmental Sciences, Peking Univ., Beijing 100871, China. – D. Dimitrov, C. Rahbek, XX and ZW, Center for Macroecology, Evolution and Climate, Natural History Museum of Denmark, Univ. of Copenhagen, Universitetsparken 15, DK-2100 Copenhagen Ø, Denmark. DD also at: Natural History Museum, Univ. of Oslo, PO Box 1172 Blindern, NO-0318 Oslo, Norway. CR also at: Imperial College London, Grand Challenges in Ecosystems and the Environment Initiative, Silwood Park Campus, Ascot, Berkshire, SL5 7PY, UK.

NCBIminer is freely available, cross-platform and user-friendly software for mining nucleotide sequence data from GenBank. It has several features that enable users to accurately and efficiently download sequences with specific attributes from the GenBank database: 1) it uses a novel search strategy, and can download sequences for distantly related taxonomic groups with high accuracy; 2) it deals with genes, CDS, rRNA, and other GenBank-defined feature types; 3) it can filter sequences by length and similarities with the reference sequence using user-defined parameters; 4) it can download information on DNA sample collections, e.g. voucher specimen, country, latitude and longitude, and collector; 5) it takes advantage of parallelization for a high efficiency workflow. We demonstrate the use and performance of NCBIminer by downloading sequences for the plant family Campanulaceaes. Compared to other methods, NCBIminer harvests more and longer sequences, and is less sensitive to query sequences.

There is a broad consensus that analyzing the phylogenetic relationships between species provides a useful lens for addressing core questions in evolutionary biology and ecology (Holt et al. 2013, Zanne et al. 2014). With the development of DNA amplification and sequencing methods and the consequent boom in studies based on molecular data, a great number of diverse genetic data from different organisms and localities have been deposited in GenBank. Databases such as GenBank have thus grown in size and complexity, requiring new tools for searching and downloading data for the generation of phylogenetic supermatrices from multiple gene sequences.

Accessibility of metadata from public DNA databases has been identified as an impediment for the development of large scale phylogenies and related research (Beaumont et al. 2005). Therefore, applications for data mining of GenBank and similar resources are becoming increasingly necessary for phylogenetic research. Currently, several standalone applications for automated sequences downloading and phylogenetic reconstruction based on either sequence similarity or GenBank annotations exist, e.g. PHLAWD (Smith et al. 2009), iPHY (Jones et al. 2011), PhyLoTA browser (Sanderson et al. 2008). GenBank annotations are not always accurate, and algorithms based only on sequence similarities will tend to reduce the lengths of output sequences. Moreover, some of these programs require above average knowledge of system administration and powerful computational resources to harvest and analyze large datasets from GenBank.

Here we developed a new program referred to as NCBIminer (NCBI data miner, available at: <http://code.google.com/p/ncbiminer/>) which is written as MATLAB scripts and compiled to a standalone application. NCBIminer is a user-friendly application for downloading sequences efficiently and automatically from GenBank with high flexibility and high accuracy. NCBIminer uses an iterated BLAST (basic local alignment search tool) algorithm to get sequences with high accuracy. The design of this application is based on currently accepted knowledge: 1) homologs of the same gene in different organisms have similar sequences or conserved domains (i.e. gene homology); 2) the more related the taxa are, the more similar are their genes. These two premises provide the foundation for NCBIminer to download a complete assemblage of sequences for a target taxonomic group with high accuracy.

Work flow

Because BLAST is based on local alignments, a BLAST search using a single query sequence usually finds only short segments or in some cases cannot find any significantly similar sequences. Normal BLAST processes often miss sequences that should be in the target but are not similar enough to the original query sequence, especially when the query sequence is phylogenetically distant from the target taxon. In order to get sequences as complete as possible for a specific locus of the target taxa, the algorithm of our application includes two steps of BLAST search.

Step 1

The first step of NCBIminer is the core of the application: select the best reference sequences (i.e. target queries) of a specific locus for a taxonomic group (e.g. a family, thereafter *Ti*) which has higher (or sometimes the same) taxonomic rank than the target taxa. The search for reference sequences in this step is seeded by the sequences (initial queries) that the user provides in the input file. Good reference sequences should be able to reflect the sequence variation of the locus across *Ti,* so that the subsequent BLAST searches using these reference sequences can catch sequence variation within *Ti*. Searching for reference sequences is particularly important when a user's taxonomic scope is wide: in such cases we also recommend providing several sequences in the input file to be used in Step 1.

Because of gene homology, it is possible to find good reference sequences of a locus for *Ti* even from phylogenetically distant initial query sequence using an iterated BLAST process. To get the reference sequences of *Ti*, the following two processes are conducted. a) Initial BLAST search based on initial query(ies): in this stage each sequence is BLASTed individually until significant BLAST hits (i.e. the part of a sequence found by BLAST) are returned for *Ti* (Fig. 1). b) These BLAST hits are subjected to the reference-sequences-identification function that we developed (see next paragraph) to identify reference sequences for *Ti*. Ideally these reference sequences should reflect the genetic variation of the required locus across entire *Ti*. These processes (a and b) will be iterated once by extracting the longest reference sequences to be used as input for BLAST searches in Step 2. The longest reference sequence normally contains more variations than short ones, and is more related to *Ti* than initial query sequences because these iterated BLASTs are constrained to *Ti*.

To find reference sequences that reflect the genetic variation of *Ti*, we developed a reference-sequences-identification function that will 1) calculate the pairwise similarities of the BLAST hits retrieved from process a), 2) group them into clusters using a hierarchical clustering method with complete linkage, and 3) choose one sequence from each cluster as a reference sequence that is then used in Step 2. BLAST hits with closer start and end match positions on the initial query sequence are more similar to each other, hence, the Euclidean distance of start and end match positions for a given pair of BLAST hits can be used to represent their similarity. For two hits, let D_1 represent their mismatches of start positions and D_2 mismatches of end positions. The distance between the two hits is calculated as $d = (D_1^2 + D_2^2)^{1/2}$ (Eq. 1). A matrix of d between all BLAST hits is then calculated and the dendrogram of these hits is generated using this matrix and the hierarchical clustering method.

The proportion (D_{diff}) of the mismatched base pairs of a pair of BLAST hits to the length of query sequence (L) can be estimated as $D_{diff} = 0.5 \times (D_1 + D_2)/L$ (Eq. 2). D_{diff} ranges from 0 to 1, and reflects the difference between the two hits relative to the query sequence. For a given d, D_{diff} reaches the maximum when $D_1 = D_2$. Then $D_1 = D_2 = L \times D_{diff}$ according to Eq. 2, and Eq. 1 can be written as $d = 2^{0.5} \times L \times D_{diff}$. To group all BLAST hits into an operational number of clusters, a reasonable threshold of d (d_0) is selected to cut the dendrogram by choosing a value of D_{diff}. The default value for D_{diff} in NCBIminer is set to 0.3, which means that the mismatches of each pair of hits within a cluster is less than 30% of the query length. This value works well for com-

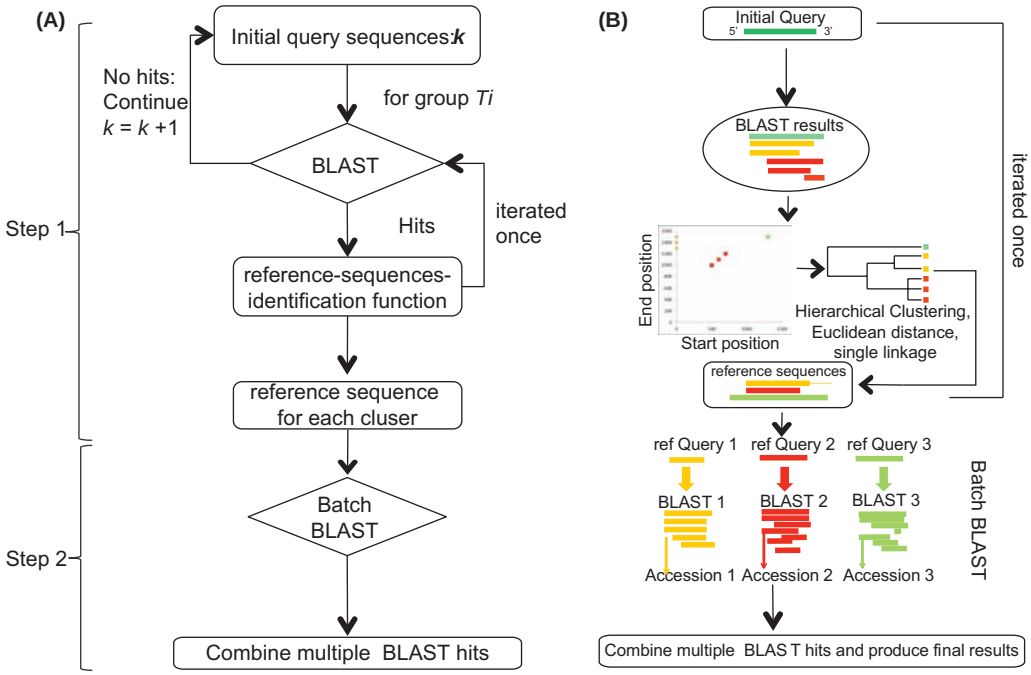

Figure 1. (A) Workflow diagram NCBIminer. (B) Graphical explanation of the two searching steps in NCBIminer algorithm. In the first, the program will use the initial query sequences to perform a BLAST search and find better reference queries for the taxon level 1 as defined by users. If hits are found, the user-defined feature type and locus names will be also used at that step. This step will be repeated once to refine the reference queries. In the second step, batch BLASTs are conducted by using the refined reference queries and the results of all queries are combined.

monly used loci including variable loci such as ITS. In general, we suggest D_{diff} = 1 for conservative genes and D_{diff} = 0.3 – 1 for variable genes. A smaller D_{diff} will generate more reference sequences and hence more BLAST runs in Step 2, which will increase the calculation time exponentially.

After cutting the dendrogram, NCBIminer will read the longest hit of each resulting cluster from GenBank according its annotation and compare it with the initial query sequence. If this sequence has a high similarity with the initial query sequence, it will be used as a reference sequence. Otherwise NCBIminer will read the next longest one until the best reference sequence for the cluster is found. To avoid possible GenBank annotation errors on start and end positions of a locus, the length of each extracted reference sequence will be adjusted according to its match with the initial query sequence used in BLAST.

Step 2

In this step, NCBIminer uses the reference sequences from Step 1 to perform BLAST searches for the target taxa, combines the hits of all BLAST searches and downloads the sequences from GenBank. BLAST hits that are found using different reference sequences may include different sequences and, in many cases, different parts of the same sequence. To optimize the final BLAST results in terms of both sequence length and the number of sequences, NCBIminer combines the BLAST hits of different reference sequences used to query the database. The combining process will 1) put together all hits with different GenBank accession numbers and 2) integrate short hits that are extracted from different parts of the same sequence into a longer one by checking their positions on the original sequence and the strand direction. In the final result, sequences with the desired minimum length (default value of 50 bps) are kept and downloaded from GenBank. Sequence direction is checked and adjusted when necessary.

In the development of NCBIminer, two MATLAB toolboxes are used: bioinformatics and parallel computing toolboxes. In NCBIminer, BLAST searches are run with the 'blastn' algorithm. Default values of BLAST parameters are also used for default NCBIminer runs except for two parameters, 'expectation value' (i.e. expected number of chance matches in a random model) and 'number of returned hits'. Default value for the former is set to 1.0×10^{-10} and that for the later is set to 5000. All BLAST parameters can be found in the input file and can be modified by users. To accelerate the computation speed, sequence downloading and parsing can be run in parallel.

Comparison with other programs

We compared NCBIminer with several existing standalone tools as shown in Table 1. Compared with these standalone programs, NCBIminer has several distinctive features. First, NCBIminer will download a complete assemblage of sequences for the requested taxonomic group using the novel two-step search strategy described above. This is helpful in providing best coverage of sequences for each species or genus in phylogeny construction and also could be useful in studies of phylogeography and population genetics. Moreover, other information of the downloaded sequences, including voucher specimen, country sampled, latitude and longitude of the sample, and collector names, could also be downloaded according to user requirements. Second, NCBIminer would compare BLAST results with GenBank annotations to avoid potential annotation errors. Finally, NCBIminer is a standalone application based on free

Table 1. Features of NCBIminer and other existing programs.

Features \ Programs	NCBIminer	PHLAWD	iPhy	PhyLoTA browser	PhyloGenerator
Operating System	Windows, Mac, Linux	Linux, Mac*	Linux, Mac*	Online service <http://phylota.net/>	Windows, Mac, Linux
Sequence parse method	Iterated BLAST; comparison between BLAST output and GenBank Annotation	BLAST	GenBank annotation for annotated sequences; BLAST similarity for unannotated sequences	All-on-all BLAST; single-linkage clustering	GenBank annotation, alignment similarity to reference sequence using length as criterion
Sequence download	All sequences for given taxonomic groups with required length	Above a user-defined coverage and identity threshold	Consensus sequences	Sequences with length > 50% of query sequences	Few different options (longest one for each species, target length, etc.)
Aim	Sequence search and download and DNA dataset construction	Phylogeny and DNA datasets construction	Phylogeny construction	Track progress on data availability; approximate alignments and phylogenies	Time-calibrated phylogenetic tree construction
Pre-installed software or pre-downloaded data	MATLAB Compiler Runtime	Python, local copy of the GenBank Database	Java, Grail, PostgreSQL, NCBI Taxonomy, BLAST, MUSCLE, EMBOSS, CAP3 etc.		
Computation power	Mainly requires internet speed	Runs locally; heavy computation power is needed	Runs locally on Linux server; heavy computation power is needed	Mainly requires internet speed	Depending on tree size and internet speed
Reference		Smith et al. (2009)	Jones et al. (2011)	Sanderson et al. (2008)	Pearse and Purvis (2013)

* installation on computers running Mac OX operating system may be possible using the Darwin development environment.

Table 2. The number of sequences of six common loci downloaded from GenBank using different methods. BLAST1 and NCBIminer1 were run with sequences of family Campanulaceae as initial query sequences, while BLAST2 and NCBIminer2 were run with sequences of a distantly-related family, Nymphaceae, as initial query sequences. Same blast parameters were used for all BLAST and NCBIminer runs. Sequences with length > 50 base pairs were kept. The numbers of erroneously-downloaded sequences were showed in bold within brackets. For NCBIminer runs, the execution time is shown within square brackets in seconds. Note that all BLAST searches are actually run on the NCBI servers and time interval between two BLAST submits is set to be 30 s to reduce the load of NCBI servers.

Feature name	Correct	GenBank annotation	BLAST1	BLAST2	NCBIminer1	NCBIminer2
atpB	576	370	576	576	576 [62 s]	576 [624 s]
atpB-rbcL spacer	403	313	403	397	403 [817 s]	403 [432 s]
rbcL	934	930	933	933	933 [246 s]	933 [464 s]
ITS1*	1198	1196 (**2**)	1097	0	1197 [517 s]	1196 [1651 s]
5.8S rRNA	926	646	919	916	919 [59 s]	919 [484 s]
ITS2*	1221	1222 (**3**)	1216	4	1220 [172 s]	1221 [441 s]

*ITS1 and ITS2 regions are difficult for BLAST. To improve the speed accuracy of BLAST, the reference sequences for ITS1 included 25 bps from the 18S rRNA and 25 bps from 5.8S rRNA and those for ITS2 indcluded 25 bps from 5.8S rRNA and 25 bps from 25S rRNA.
*As the heaviest work in NCBIminer, i.e. BLAST searching, is done in the NCBI servers, execution time depends on internet speed, how busy the NCBI BLAST server is, how many BLAST runs are submitted and the variability of require loci. It takes longer for variable than for conservative loci.

MATLAB Compiler Runtime (MCR, <www.mathworks.cn/products/compiler/mcr/>) and does not require other extra software or programming experience for the user. The advantage of NCBIminer is obvious when compared with other basic sequence downloading functions available in different programing languages (e.g. perl, R, ruby, biopython, eautils, MATLAB): none of these functions uses the search strategy described here and they do require much more programming experience.

Examples and evaluation

Here we use the family Campanulaceae as an example to evaluate the capability of NCBIminer to accurately download sequences for a specified taxon. To run NCBIminer, users simply need to prepare a text file as input file. An example of a generic input file and the file used for this evaluation are distributed with the package.

For Campanulaceae, we first manually downloaded all sequences of atpB, atpB-rbcL spacer, rbcL, ITS1, 5.8S rRNA, ITS2 (searched on 1 June) and aligned them using mafft ver. 7.158 (Katoh and Standley 2013). All aligned sequences were checked manually. In total, 2855 accessions were found for these six loci (Table 2). These carefully checked sequences were used as references to evaluate the results from different sequence downloading methods. Here the accuracy of downloaded sequences was evaluated by the identity and mismatches between them and the reference set of sequences. For a downloaded sequence *a* and its corresponding reference sequence *b*, let L_a, L_b and L_c represent the lengths of *a*, *b* and the matched base pairs between a and b,

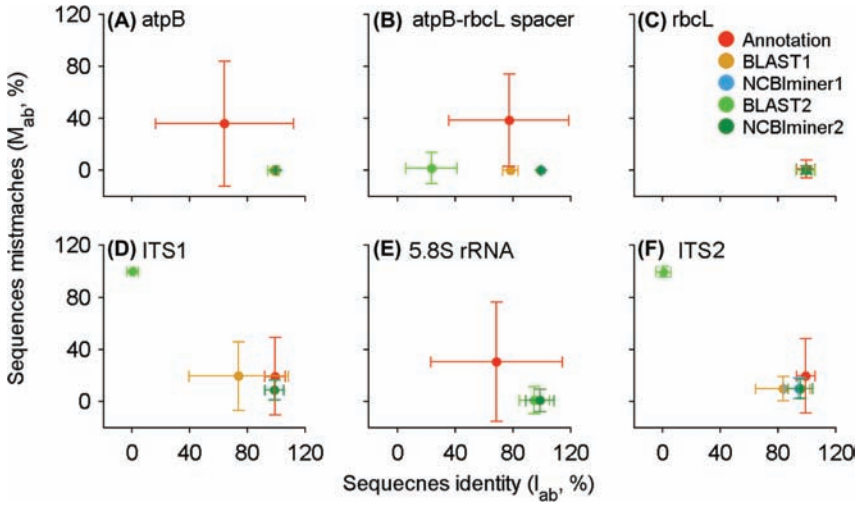

Figure 2. Accuracy comparisons among different methods for six loci of family Campanulaceae. Identity (I_{ab}) and mismatches (M_{ab}) between the reference sequences and the sequences downloaded were calculated for each method separately. See text for the definition of I_{ab} and M_{ab}. The x axes show the mean I_{ab} of all downloaded sequences, and the y axes show the mean M_{ab}. The bars at each dot are corresponding standard deviations of I_{ab} and M_{ab}. NCBIminer 1 and 2 showed no difference and always fell at the right bottom corner, suggesting that NCBIminer works better than the other methods and its performance is not influenced by initial query sequences and loci. Annotation: GenBank annotation; NCBIminer1 and BALST1: NCBIminer run and normal BLAST search with sequences of Campanulaceae as query sequences respectively; NCBIminer2 and BLAST2: NCBIminer run and normal BLAST search with sequences of a distantly-related family (Nymphaeaceae) as query sequences.

respectively. Sequence identity is defined as $I_{ab} = L_c/L_b$, and sequence mismatches is defined as $M_{ab} = (L_a - L_c)/L_a$. If a is equal to b, I_{ab} is 1 and M_{ab} is 0. With the increase of the difference between a and b, I_{ab} decreases and M_{ab} increases. When a is completely different from b, I_{ab} becomes 0 and M_{ab} becomes 1.

Using I_{ab} and M_{ab} we compared the accuracy of NCBIminer with normal BLAST search and sequence search using GenBank annotations, which are both widely used for downloading sequences from GenBank. NCBIminer and normal BLAST searches were both conducted twice using different initial query sequences in order to check if these two methods are sensitive to the selection of initial query sequences. The first run (NCBIminer1, BLAST1) used sequences of Campanulaceae species as initial query sequences while the second run (NCBIminer2, BLAST2) used sequences of a distantly related family, Nymphaeaceae. In NCBIminer, taxon levels 1 and 2 were both Campanulaceae. For variable genes (including atpB-rbcL spacer, ITS1 and ITS2), expectation value was 0.001, and D_{diff} was 0.3. For conservative genes (including aptB, rbcL and 5.8S rRNA), expectation value was 10^{-10}, and D_{diff} was 1. Default values were used for other parameters. The same BLAST parameters were used in the normal BLAST searches. NCBIminer runs on a 64-bit Windows 7 computer with 12 cores in Peking Univ.

The two NCBIminer runs both downloaded all sequences for the six loci with high accuracy (Table 2). The average I_{ab} of all downloaded sequences was close to 1 and average M_{ab} close to 0 (Fig. 2). The standard deviations of I_{ab} and M_{ab} for NCBIminer runs were small (Fig. 2). In contrast, the GenBank annotation method downloaded fewer sequences than NCBIminer (Table 2). It is noteworthy that the standard deviations of I_{ab} and M_{ab} of the sequences downloaded by GenBank annotation were large, which suggests that GenBank annotation is not always reliable. Two sequences were erroneously annotated as ITS1 and 3 as ITS2 in GenBank. For variable loci like ITS1 and ITS2, the normal BLAST method was very sensitive to the selection of initial query sequences (Table 1). Specifically, BLAST1 using Campanulaceae sequences as reference sequences performs as well as NCBIminer1, while BLAST2 using sequences of a distantly-related family as reference sequences downloaded only four sequences for ITS2 and none for ITS1. Our results indicate that NCBIminer has higher accuracy in downloading sequences from GenBank than GenBank annotation and normal BLAST. More importantly, the output of NCBIminer is not influenced by the selection of the initial query sequence.

To cite NCBIminer or acknowledge its use, cite this Software note as follows, substituting the version of the application that you used for 'version 1.0':

Xu, X., Dimitrov, D., Rahbek, C. and Wang, Z. 2015. NCBIminer: sequences harvest from Genbank. – Ecography 38: 426–430 (ver. 1.0).

Acknowledgements – We thank Dan Flynn for his help with writing, and Weihua Du for program debugging. XX was supported by Chinese Scholarship Council, and ZW by the Marie Curie Actions (PIEF-GA-2010-275666). All authors thank the Danish National Research Foundation for support to the Center for Macroecology, Evolution and Climate. This study was supported by the National Natural Science Foundation of China (31470564, 31400467, 31321061), and the 111 Project (B14001).

References

Beaumont, L. J. et al. 2005. Predicting species distributions: use of climatic parameters in BIOCLIM and its impact on predictions of species' current and future distributions. – Ecol. Model. 186: 251–270.

Holt, B. G. et al. 2013. An update of Wallace's zoogeographic regions of the world. – Science 339: 74–78.

Jones, M. et al. 2011. iPhy: an integrated phylogenetic workbench for supermatrix analyses. – BMC Bioinform. 12: 30.

Katoh, K. and Standley, D. M. 2013. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. – Mol. Biol. Evol. 30: 772–780.

Pearse, W. D. and Purvis, A. 2013. phyloGenerator: an automated phylogeny generation tool for ecologists. – Methods Ecol. Evol. 4: 692–698.

Sanderson, M. J. et al. 2008. The PhyLoTA browser: processing GenBank for molecular phylogenetics research. – Syst. Biol. 57: 335–346.

Smith, S. et al. 2009. Mega-phylogeny approach for comparative biology: an alternative to supertree and supermatrix approaches. – BMC Evol. Biol. 9: 37.

Zanne, A. E. et al. 2014. Three keys to the radiation of angiosperms into freezing environments. – Nature 506: 89–92.