PDA - Phylogenetic Diversity Analyzer

PDA Manual

Version 1.0 (August 2014)

Core developers:

Bui Quang Minh - `minh.bui(at)mfpl.ac.at`
Olga Chernomor - `olga.chernomor(at)mfpl.ac.at`

Support:

Arndt von Haeseler - `arndt.von.haeseler(at)mfpl.ac.at`
Steffen Klaere - `steffen.klaere(at)gmail.com`

Contact address:

Center for Integrative Bioinformatics Vienna (CIBIV)
Max F. Perutz Laboratories, University of Vienna, Medical University of Vienna
Dr. Bohr-Gasse 9, A-1030 Vienna, Austria

Contents

1 Introduction

Phylogenetic Diversity (PD), coined by Faith (1992), is a quantitative measure to assess the biodiversity of species based on a phylogeny. Given n taxa (or species) connected by a phylogenetic tree, the PD of any subset of taxa is defined as the sum of the branch-lengths of the minimal subtree connecting these taxa. We have recently proposed *Split Diversity (SD)* as an extension of PD for split networks (or split systems). Given a split system of n taxa, the SD of any subset of taxa is equal to the sum of the weights of all the splits separating at least two taxa in the subset. This definition coincides with the PD when the underlying split system corresponds to a tree.

Phylogenetic Diversity Analyzer (PDA) provides a wide range of biodiversity analysis using *Phylogenetic Diversity (PD), Split Diversity (SD)* and related measures based on both phylogenetic trees and networks. This provides conservation biologists with an objective decision-making process. The major features include:

- Maximizing PD and SD given various types of constraints including budgetary, geographical, and ecological constraints.
- Minimizing budget given diversity threshold.
- Evaluation of predefined sets of taxa (e.g. in an area) including exclusive, endemic, and complementary PD/SD.

PDA was previously named "Phylogenetic Diversity Algorithm". Since version 0.5 we decided to change it to *Phylogenetic Diversity Analyzer* due to its extended functionalities. PDA is available free of charge under GNU GPL license from

`http://www.cibiv.at/software/pda/`

Please read the *Installation* section 2 for more details how to install the software. An easy-to-use web-interface is now available at

`http://www.cibiv.at/software/pda/web-pda/`.

We suggest that this documentation should be read before using *PDA* the first time. To find out what's new in the current version please read the *Version History* section 7.

1.1 Methods

The methods have been described in details in:

- O. Chernomor, B.Q. Minh, F. Forest, S. Klaere, T. Ingram, M. Henzinger, and A. von Haeseler (2014) Split Diversity in Constrained Conservation Prioritization using Integer Programming, submitted.

- B.Q. Minh, S. Klaere, and A. von Haeseler (2010) SDA*: A Simple and Unifying Solution to Recent Bioinformatic Chaallenges for Conservation Genetics. *Proceedings of the 2nd International Conference on Knowledge and Systems Engeneering - KSE-2010* (Hanoi, Vietnam), 33-37, IEEE Computer Society, Los Alamitos, CA, USA.
- B. Q. Minh, S. Klaere, and A. von Haeseler (2009) Taxon Selection under Split Diversity. *Syst. Biol.*, **57**, 586–594.
- B.Q. Minh, F. Pardi, S. Klaere, and A. von Haeseler (2009) Budgeted Phylogenetic Diversity on Circular Split Systems. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **6**, 22-29.
- B.Q. Minh, S. Klaere, and A. von Haeseler (2006) Phylogenetic diversity within seconds. *Syst. Biol.*, **55**, 769–773.

2 Installation

See below for information how to install/build the different versions of the PDA software. Executables are intended for a number of operating systems.

Note, that in order to use methods based on Integer Linear Programming you need to install Gurobi Optimizer (Gurobi Optimization Inc., 2013). Gurobi is free for academic use and when it is installed on your computer, PDA will call it automatically.

2.1 Binary release

1. Download the executable version of PDA for your operating system if it is available (`pda-XXX-OS.tar.gz` or `pda-XXX-OS.zip`, where `XXX` is the current version number and OS the operating system) from
`http://www.cibiv.at/software/pda`
2. Extract the files (e.g., with `tar xvzf pda-XXX-OS.tar.gz` under Unix). This should create a directory `pda-XXX-OS`.
3. You will find the executable in `pda-XXX-OS/`. This executable you should rename to `pda` (or `pda.exe` on Windows systems) and copy it to your system search path such that it is found by your system.

If you encounter problems, please ask your local administrator for help.

3 Command-line options

If you run 'pda -h', PDA will display a usage screen. The meanings of the options are mainly what you see (For explanations and possible usages see the subsequent sections).

```
Usage: pda [OPTIONS] <file_name> [<output_file>]
GENERAL OPTIONS:
  -h                Print this help dialog.
  <file_name>       Input tree in NEWICK/NEXUS format or network in NEXUS format
  <output_file>     Output file to store results, default is '<file_name>.pda'
  -k <num>          Find optimal set of size <num>
  -k <min>:<max>    Find optimal sets of size from <min> to <max>
  -k <min>:<max>:<step>
                    Find optimal sets of size min, min+step, min+2*step,...
  -k% <k_percent>   Find optimal set of size in percentage
  -o <taxon>        Root name to compute rooted PD (default: unrooted)
  -if <file>        File containing taxa to be included into optimal sets
  -e <file>         File containing branch/split scale and taxa weights
  -all              Identify all multiple optimal sets
  -lim <max_limit>  The maximum number of optimal sets for each k if -a is specified
  -min              Compute minimal sets (default: maximal)
  -1out             Print taxa sets and scores to separate files
  -oldout           Print output compatible with version 0.3
  -v                Verbose mode

OPTIONS FOR PHYLOGENETIC DIVERSITY (PD):
  -root             Make the tree ROOTED, default is unrooted
    NOTE: this option and -o <taxon> cannot be both specified
  -g                Run greedy algorithm only (default: auto)
  -pr               Run pruning algorithm only (default: auto)

OPTIONS FOR BUDGET CONSTRAINTS:
  -u <file>         File containing total budget and taxa preservation costs
  -b <budget>       Total budget to conserve taxa
  -b <min>:<max>    Find all sets with budget from <min> to <max>
  -b <min>:<max>:<step>
                    Find optimal sets with budget min, min+step, min+2*step,...

OPTIONS FOR AREA ANALYSIS:
  -ts <area_file>   Compute/maximize PD/SD of areas (combine with -k to maximize)
  -excl             Compute exclusive PD/SD
  -endem            Compute endemic PD/SD
  -compl <areas>    Compute complementary PD/SD given the listed <areas>

OPTIONS FOR VIABILITY CONSTRAINTS:
  -eco <food_web>   File containing food web matrix
  -diet <min_diet>  Minimum diet portion (%) to be preserved for each predator

MISCELLANEOUS:
  -dd <sample_size> Compute PD distribution of random sets of size k
```

3.1 General options

- `<file_name>` option

  The `<file_name>` will be the input tree file in NEWICK format or input split network in NEXUS format. The only exception is when you set `-r[u] <num_taxa>`, the program will generate a random tree and write it into the `<file_name>` file.

  More information on NEWICK tree format can be found at `http://evolution.genetics.washington.`

  More information on NEXUS file format can be found in the article Maddison *et al.* (1997) or at `http://awcmee.massey.ac.nz/spectronet/nexus.html`.

- `<output_file>` option

  This is to set the output file name instead of the default `<file_name>.pda` where `<file_name>` is the input file name defined above.

- `-k <num_taxa>`, `-k <min>:<max>`, and `-k <min>:<max>:<step>` option

  With `-k <num_taxa>`, PDA will compute the optimal *PD* sets of size `<num_taxa>`. With the new option `-k <min>:<max>`, PDA will compute the optimal PD_k sets for k from `<min>` to `<max>`. So you do not have to run PDA several times on the same tree or network for different k, and thus save a lot of computational time. It is even more convenient in some case with `-k <min>:<max>:<step>`: PDA will only report the optimal *PD* sets of size k from from `<min>` to `<max>` with the a jumping-step of `<step>`. That means k will iterate through `<min>`, `<min>+<step>`, `<min>+2*<step>`,... until not exceeding `<max>`.

- `-k% <k_percent>` option

  This option is similar to `-k <num>`, but defines the subset size as a percentage of the total number of taxa or areas.

- `-o <taxon>` option

  From version 0.3, one can distinguish between unrooted and rooted *PD* by this option. See Faith and Baker (2006) for a discussion. If your tree/network has a specific root or outgroup, always specify it by this option. The root will then be always included into the final *PD* set.

- `-e <file>` option

  The `<file>` must be in the following format:

  1. First line contains a scaling factor for every branch length/split weight in the tree/network.
  2. Second to last line: each line contains a taxon name and its taxon weight (the "importance" of that taxon). Any taxa not listed here will be assigned a taxon weight of ZERO. If you prefer some taxa, you can give them a positive taxon weight. Specify a very high taxon weight to your "favourite" taxa if you want to always include them into your final optimal *PD* set.

Then the input tree/network will be processed as follows: all branch lengths/split weights will be multiplied with the scaling factor, and then all external branch lengths/trivial split weights will be increased with the specified taxon weights. This processed tree/network will replace the input tree/network for the analysis.

More information on those additional parameters can be found in Steel (2005).

- `-if <file>` option

  The file containing all taxon names, which you want to always include into your final PD-set irrespective of any constraints. The format is simply a list of all taxon names separated by blank(s) or new line. An error will be displayed if some taxon name does not appear in the input tree/network.

  This option might be handy in comparative genomics when you have already sequenced several species and have to make a decision what species to be sequenced next. Then the species names, which were already sequenced, can be listed in this file. See Pardi and Goldman (2005) for a discussion.

- `-a, -all` option

  This option allows you to identify multiple optimal PD_k sets for a specific k. This is useful in case there are more than one optimal sets with the same PD score. Note that if you specify `-k <min>:<max>`, PDA will handle correctly for each k. The `-a` option can be used in conjunction with `-lim <max_limit>` option.

- `-lim <max_limit>` option

  When you set `-a` option, the number of multiple optimal PD_k sets for each k to be reported will be limited to at most `<max_limit>`. The default limit is 100 if you don't specify. This is simply to avoid PDA from memory overflow if millions of such optimal sets exist.

- `-min` option

  This option tells PDA to find the minimal PD sets instead of the default maximal ones. Note that algorithmically, on trees the greedy algorithm does not work anymore for PD minimization. However, the dynamic programming algorithm presented in Minh *et al.* (2009) can be easily adapted for this case by negating all the branch lengths!

- `-1out` option

  This tells PDA to write the list of taxa sets into `*.pdtaxa` file and the PD scores into `*.score` file.

- `-oldout` option

  This is for compatibility reason only since by default, version 0.5 only produces the output file `*.pda` which contains more information than only optimal sets, scores, and sub-trees. So this option tells PDA to write extra resulting files as outputed in version 0.3.

- `-v` option

  This option tells PDA to print more intermediate information while running.

3.2 Options for budget constraints

From version 0.5 PDA is extended to cope with budget constraints. The extended problem is formulated as follows. Given a tree or split network, integer preservation costs c_s for each taxon s and a total integer budget B. Find a subset S of taxa to maximize $PD(S)$ such that the total cost do not exceed the given budget: $\sum_{s \in S} c_s \leq B$. The restriction to integer numbers is not limitation since budgets are normally expressed in integer. If not, it can be easily transformed into integer. This problem is not solvable by a greedy strategy but by a dynamic programming algorithm. A paper describing this is still in preparation.

- `-u <file>` option

  This file is in the following format. The first line contains the total integer budget. Each of the subsequent lines contains a taxon name and an associated integer cost, separated by blank(s). Note that any taxon which are not given a cost will be automatically assigned a cost of ZERO.

- `-b <budget>` option

  If you don't want to use the budget written in the file specified by `-u <file>` option, use this option. The budget specified here will be taken for laler analysis.

- `-b <min>:<max>[:<step>]` option

  This has the same effect as described for `-k <min>:<max>[:<step>]` option (see Section 3.1).

3.3 Options for area analysis

PDA is capable of computing and maximizing the PD/SD scores of areas. An area simply refers to a user-defined subset of taxa. It also can compute the exclusive, endemic, and complementary PD/SD of areas. In the following we only describe PD for the sake of simplicity, but all options work with SD as well.

- `-ts <area_file>` option

  The list of areas is given in `<area_file>`. This file can be in one of the two formats: simple text file or NEXUS format. PDA will automatically detect the type of this file.

  - Simple text file: The first line is the number of taxa n_1 of the first area. The next n_1 lines are the names of n_1 taxa in the first area. Then comes the number n_2 (the number of taxa of the second area) and n_2 lines storing the names of those taxa in the second area. This repeats until the last area or reaching the end of file.
  - NEXUS format: PDA will read the SETS block of the NEXUS file. For simplicity, here is an example:

```
#nexus

begin sets;
    taxset 'a1' = 'a' 'b' 'c';
    taxset a2 = c d g n;
    taxset a3 = h i;
    taxset a4 = j k;
    taxset a5 = l m;
    taxset a6 = h i;
    taxset a7 = j k;
    taxset a8 = l m;
end; [sets]
```

- `-k <num_area>` option
  This tells PDA to compute the maximal PD set of areas.

- `-excl` option

  This tells PDA to compute the exclusive PD of all areas as well. In short, given X the set of all taxa and A an area, the exclusive PD of A is simply: $ePD(A) = PD(X) - PD(X - A)$. See Lewis and Lewis (2005) for the original description of this measure.

- `-endem` option

  This tells PDA to compute the endemic PD of all areas as well. Given X the set of all taxa and $A_1, A_2, \ldots, A_m$ all areas you have. Let U be the union taxon set of all areas. Then the endemic PD of a particular area A_i is: $PD(A_1 \cup \ldots \cup A_m) - PD(A_1 \cup \ldots A_{i-1} \cup A_{i+1} \cup \ldots \cup A_m)$. See Faith *et al.* (2004) for more details and interpretation.

- `-compl <areas>` option

  This tells PDA to compute the PD-complementarity of all areas given the list `<areas>`. The list can contain one area name or several area names separated by commas. Let B the union taxon set of all given areas. Then the PD-complementarity of a particular area A_i is: $PD(A_i|B) = PD(A_i \cup B) - PD(B)$. See Faith *et al.* (2004) for more details and interpretation.

3.4 Options for viability constraints NEW

In case when the taxon sets from the tree/split network and from the food web are not equal, the analysis will continue with the union set. Those taxa not present in the food web won't be constrained by viability. However, it is advised to use the tree/split network and food web, which do not differ much in the species composition. General options which can be used in combination with viability constrained analysis:

```
<file_name>       User tree in NEWICK format or split network in NEXUS format
<output_file>     Output file to store results, default is '<file_name>.pda'
-k <num_taxa>     Find optimal set of size <num_taxa>
-o <taxon>        Root name to compute rooted PD, default is unrooted
-if <file>        File containing taxa to be included into optimal set
-v                Verbose mode
```

and the following are the additional options.

- `-eco <food web file>` option

  This file contains the food web matrix. The first line specifies the number N of taxa in the food web. Each next line starts with a taxon name followed by N matrix entries. Each matrix entry $w_{ij} \geq 0$ defines the portion of prey i in diet of predator j. The matrix can also be defined just by 0 or 1, meaning that taxon i is not a prey or a prey of predator j respectively. Important! PDA supports only acyclic food webs. In case of cannibalism ($w_{ii} \neq 0$) PDA will set the entry to 0 and continue with the processed food web. However, if there are still cycles, PDA prints the message and stops.

  ```
  Example of a food web file:
  5
  taxon_name_1 0 0 0 0 0
  taxon_name_2 1 0 0 0 0
  taxon_name_3 1 0 0 0 0
  taxon_name_4 0 1 1 0 0
  taxon_name_5 1 0 1 0 0
  ```

- `-diet <% diet>` option

  This option specifies the minimum diet portion to be preserved for each predator. Skipping it or using 0 results in the naive viability constraint. For diet greater than 0, the d%-viability constraint is used.

4 Outputs

All outputs will be written to `<file_name>.pda` by default or `<output_file>` if you specify it in the command-line.

If you specify `-1out`, all the taxa sets are additionally written to `<file_name>.pdtaxa` and the *PD* scores are written to `<file_name>.score`.

If you specify `-oldout`, additional files are written as of version 0.3 as follows.

Resulting *PD* taxa set will be written into: `<file_name>.<k>.pdtaxa`

If the option `-a` or `-all` is specified and multiple optimal is observed, subsequent optimal taxa sets will be written into: `<file_name>.<k>.pdtaxa.1`,
`<file_name>.<k>.pdtaxa.2`,...

The score is printed to `<file_name>.score`.

For tree, resulting sub-trees are written into:

- If you specify `-b` or `--both`:
  `<file_name>.<k>.greedy` for greedy algorithm, and
  `<file_name>.<k>.pruning` for pruning algorithm.
- Otherwise: `<user_tree>.<k>.pdtree`.

In case when viability constraints are used PDA outputs a food web restricted to the optimal subset of taxa into file `<food_wed.subFoodWeb>`.

NOTE:

- Two options `-1out` and `-oldout` are mutually exclusive. That means if you specify both, the later specified option in the command-line will override the earlier one.
- For the case of split network, no resulting sub-network is written.
- If you choose option to generate a random tree/network, it will be written to `<file_name>` (it acts as output instead of the input file).

5 Example usages

5.1 Example usages for trees

```
./pda test.tree -k 4
```

Infer the maximal *PD*-tree of 4 taxa from the tree in `test.tree` (in NEWICK format). *gPDA* or *pPDA* algorithm (Minh *et al.*, 2006) will be determined automatically. Resulting tree will be written to `test.tree.4.pdtree`. Resulting taxa set will be printed to `test.tree.4.pdtaxa`.

NOTE: The program will automatically detect the type of the input file (either NEWICK or NEXUS) to apply appropriate *PDA* algorithms. It should not depend on the file name (`.tree` or `.nex` does not matter).

```
./pda test.tree -k 4 -o c
```

Compute the "rooted *PD*", the tree is rooted at taxon `'c'`. `'c'` will be included into the final *PD*-set.

```
./pda test.tree -k 4 -g
```

Same as the first command, but only apply the $gPDA$ algorithm.

```
./pda test.tree -k 4 -b
```

Run both algorithms. Resulting trees will be written into
`test.tree.4.greedy` and `test.tree.4.pruning`.

```
./pda test.tree -k 4 -e test.pam
```

Read the weight information from `test.pam` file and integrate this into the tree in `test.tree`. Then run the program as the first example command.

```
./pda test.tree -k 4 -i test.taxa
```

Include the "favourite" taxa listed in `test.taxa` into the final *PD*-set.

```
./pda test.tree -k 4 -e test.pam -i test.taxa
```

Combining both features of the above two example commands.

```
./pda 1000.tree -r 1000
```

Generate a 1000-taxa random tree under Yule Harding Model. Write resulting tree into `1000.tree` file under NEWICK format.

5.2 Example usages for networks

```
./pda test.nex -k 3
```

Find the maximal PD_3 set of the split network in `test.nex` (in NEXUS format, as produced by e.g., SplitsTree 4 program (Huson and Bryant, 2006)). *PDA* will detect whether the input split system is circular or not. If yes, apply the dynamic programming algorithm, otherwise, use exhaustive search. Resulting taxa set will be printed to `test.nex.3.pdtaxa`.

```
./pda test.nex -k 3 -o 2
```

Compute the "rooted *PD*", the split system is rooted at taxon `'2'`. `'2'` will be included into the final *PD*-set.

NOTE: With this option, the program will normal perform much faster (the time-complexity reduces by a factor of n, where n is the number of taxa). So always specify `-o <taxon_name>` if you are sure that some taxon must be present in the final *PD* set (e.g. the taxon with a very long terminal split).

Other basic options (`-i, -e <file>`) should also work fine with split network.

```
./pda test.nex -k 4 -mk 2
```

Identify all optimal *PD* sets containing 2 to 4 taxa. The resulting *PD* sets will be printed to `test.nex.2.pdtaxa`, `test.nex.3.pdtaxa`, `test.nex.4.pdtaxa`. The *PD* scores are written to `test.nex.score` containing several lines. Each line as
`<sub_size> <corresponding_score>`, where `<sub_size>` should go from 2 to 4.

```
./pda test.nex -k 3 -all
```

Find all multiple optimal PD_3 sets: if there are more than 1 optimal 3-set, all of them will be printed. The second optimal set will be in
`test.nex.3.pdtaxa.1`, the third in `test.nex.3.pdtaxa.2`, etc.

NOTE: This optimal might lead to exponential computing time, as it actually depends on the number of optimal *PD* sets!

```
./pda test.nex -k 4 -mk 2 -all
```

Combine the features of the two previous commands.

6 Howto employ the method for networks

A way to employ this new feature is to use together with program SplitsTree 4 (Huson and Bryant, 2006), available on the website `http://www.splitstree.org/`. First, you recontruct a circular network by e.g., Neighbor-net method (Bryant and Moulton, 2004). The resulting network is then saved to a NEXUS file, e.g., `mynet.nex`. Then you can feed `mynet.nex` directly to *PDA*.

There could be several blocks in `mynet.nex` input file. However, *PDA* only cares for `TAXA` and `SPLITS` blocks. All other will be ignored, including `CHARACTERS, DISTANCES, NETWORKS,` ST_ASSUMPTIONS, etc. You can also prepare your own split network. A simple input file is inside the src/ folder under the name test.nex.

NOTE: The algorithm for circular network is very fast. So always specify the CYCLE command inside the SPLITS block of the NEXUS file. Otherwise, an exhaustive search will be applied and very slow.

7 Version History

1.0 Extension to viability constrained analysis.

0.5.1 Some bugs fixed and codes cleaned. A new user manual.

0.5 Extension to budgeted PD. Being able to compute PD and PD-related measures for areas.

0.3 Extension to split networks. Distinguish between unrooted and rooted PD. Print also the taxa set now.

0.21 Fix a minor bug with STL vector constructor while compiling.

0.2 Inclusion of `-i <file>` option.

0.1 Initial version.

8 Credits

The parser for NEXUS file format is derived from the Nexus Class Library (Lewis, 2003).

Acknowledgement

The authors would like to thank Dan Faith for helpful suggestions on the subject and Heiko Schmidt and Tanja Gesell for fruitful discussions. We also thank Mike Steel for constructive comments. Financial support from the Wiener Wissenschafts-, Forschungs- and Technologiefonds (WWTF) and the University of Vienna are greatly appreciated.

References

Bryant, D. and Moulton, V. (2004) Neighbor-net: An agglomerative method for the construction of phylogenetic networks. *Mol. Biol. Evol.*, **21**, 255–265.

Faith, D., Reid, C. and Hunter, J. (2004) Integrating phylogenetic diversity, complementarity, and endemism for conservation assessment. *Conserv. Biol.*, **18**, 255–261.

Faith, D. P. (1992) Conservation Evaluation and Phylogenetic Diversity. *Biol. Conserv.*, **61**, 1–10.

Faith, D. P. and Baker, A. M. (2006) Phylogenetic diversity (PD) and biodiversity conservation: Some bioinformatics challenges. *Evolutionary Bioinformatics Online*, **2**, 70–77.

Gurobi Optimization Inc. (2013) Gurobi optimizer reference manual. *http://www.gurobi.com.*

Huson, D. H. and Bryant, D. (2006) Application of phylogenetic networks in evolutionary studies. *Mol. Biol. Evol.*, **23**, 254–267.

Lewis, L. A. and Lewis, P. O. (2005) Unearthing the molecular diversity of desert soil green algae. *Syst. Biol.*, **54**, 936–947.

Lewis, P. O. (2003) NCL: a C++ class library for interpreting data files in NEXUS format. *Bioinformatics*, **19**, 2330–2331.

Maddison, D. R., Swofford, D. L. and Maddison, W. P. (1997) NEXUS: An extensible file format for systematic information. *Syst. Biol.*, **46**, 590–621.

Minh, B. Q., Klaere, S. and von Haeseler, A. (2006) Phylogenetic diversity within seconds. *Syst. Biol.*, **55**, 769–773.

Minh, B. Q., Pardi, F., Klaere, S. and von Haeseler, A. (2009) Budgeted phylogenetic diversity on circular split systems. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **6**, 22–29.

Pardi, F. and Goldman, N. (2005) Species choice for comparative genomics: Being greedy works. *PLoS Genet.*, **1**, 672–675.

Steel, M. (2005) Phylogenetic diversity and the greedy algorithm. *Syst. Biol.*, **54**, 527–529.