SOFTWARE TOOL ARTICLE

pTITAN2: Permutation of treatment labels and Threshold Indicator Taxa ANalysis [version 1; peer review: 1 not approved]

Stephanie Figary ¹, Peter DeWitt ², Naomi Detenbeck ³

¹ORISE participant at U.S. Environmental Protection Agency, Cornell University, Narragansett, Rhode Island, 02882, USA

²National Renewable Energy Laboratory, Golden, CO, 80401, USA

³Atlantic Coastal Environmental Sciences Division, U.S. Environmental Protection Agency, Narragansett, RI, 02882, USA

V1 First published: 02 Mar 2022, 11:267
<https://doi.org/10.12688/f1000research.83714.1>

Latest published: 02 Mar 2022, 11:267
<https://doi.org/10.12688/f1000research.83714.1>

Abstract

Background: Taxa Indicator Threshold ANalysis (TITAN) was developed to identify thresholds along environmental gradients where rapid changes in taxa frequency and relative abundance are observed. TITAN determines separate change-points for increasing and decreasing taxa in aggregate, as well as change-points for individual taxa, with associated confidence intervals generated using bootstrapping. However, if TITAN is applied to different classes of observations, additional analyses besides using non-overlapping confidence intervals are needed to establish whether change-points differ between treatments or groups because non-overlapping confidence intervals can indicate significant differences but overlapping confidence intervals do not necessarily mean the null hypothesis cannot be rejected.

Methods: To address this, we present a new R package, pTITAN2, which is an extension to the existing TITAN2 package. The pTITAN2 package was developed to enable comparisons of TITAN output between treatments by permutating the observed data between treatments and rerunning TITAN on the permuted data.

Results: The pTITAN2 package includes two functions, occurrences and permute. The occurrences function selects the taxonomic codes to be used in a TITAN run while maintaining the most taxonomic details. The permute function is then used to create a list of permuted sets of taxa and environmental gradients. TITAN is then run again on the permuted data and p-value test can be calculated using the observed and permuted TITAN output to test for statistical differences between treatment effects.

Conclusions: The package pTITAN2 is an extension of the existing TITAN2 package and enables users to perform the appropriate statistical tests and determine statistical differences without using overlapping confidence intervals.

Open Peer Review

Approval Status

1

version 1

02 Mar 2022

[view](#)

1. Cajo J. F. ter Braak , Wageningen
University & Research, Wageningen, The
Netherlands

Any reports and responses or comments on the article can be found at the end of the article.

Keywords

TITAN, permutations, thresholds, community composition

This article is included in the **Bioinformatics** gateway.

This article is included in the **RPackage** gateway.

Corresponding author: Naomi Detenbeck (detenbeck.naomi@epa.gov)

Author roles: **Figary S:** Conceptualization, Formal Analysis, Methodology, Validation, Writing – Original Draft Preparation; **DeWitt P:** Conceptualization, Software, Writing – Review & Editing; **Detenbeck N:** Conceptualization, Methodology, Project Administration, Resources, Supervision, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work was partially supported by the U.S. Environmental Protection Agency via an interagency agreement with the Department of Energy (DW92429801-9) which provided funding to Stephanie Figary through the ORISE program and through funding on the EPA contract EP-C-13-022 with Neptune, supporting software development.

Copyright: © 2022 Figary S *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Figary S, DeWitt P and Detenbeck N. **pTITAN2: Permutation of treatment labels and Threshold Indicator Taxa ANalysis [version 1; peer review: 1 not approved]** F1000Research 2022, 11:267 <https://doi.org/10.12688/f1000research.83714.1>

First published: 02 Mar 2022, 11:267 <https://doi.org/10.12688/f1000research.83714.1>

1. Introduction

Community ecologists are interested in understanding the structure and interactions of multiple species in a given area or habitat type and many are interested in understanding how communities respond to changing environmental or anthropogenic gradients. One method a community ecologist can use for understanding or detecting changes in ecological thresholds across environmental gradients is Taxa Indicator Threshold ANalysis (TITAN) (Baker and King 2010). TITAN is useful for determining the impacts of environmental or anthropogenic gradients in community ecology studies because it both analyzes each individual taxa response and the community as a whole in the same analysis. Additionally, unlike other community ecology methods (Gauch and Gauch 1982), TITAN separates the taxa that increase from those that decrease across an environmental gradient to provide a more complete picture of the community response to that gradient.

As an overview, TITAN methods use change point (King and Richardson 2003; Qian, King, and Richardson 2003) and indicator species analysis (Dufrêne and Legendre 1997) to determine the point along an environmental gradient, such as percent impervious cover (IC) in the upstream watershed area, where individual taxa have the largest change in occurrence frequency and abundance, and then uses the individual taxa results to determine synchronous areas of taxa change at the community level (Baker and King 2010, 2013). In TITAN, individual taxa change points are determined by calculating the taxa's indicator score (IndVal) along the environmental gradient and assigning the taxa as either increasing (z+) or declining (z-) in response to an increase in the environmental gradient variable. Permutations of data in TITAN runs are used to calculate the likelihood of random data generating a larger IndVal score than the observed data and to standardize the taxa response to the environmental gradient by calculating the taxa z-scores using permuted distributions. Taxa z-scores are added together for increasing (z+) and declining (z-) taxa along the environmental gradient and the point with the highest sum (z-) and sum (z+) score is defined as the community change point. Next, bootstrapping of observed data is used to calculate percentiles around the community and individual taxa z-scores, and to determine if individual taxon responses are pure and reliable. Purity is a measure of the proportion of bootstrap results that match the taxa's observed response group as either increasing or declining. Reliability is the proportion of bootstraps with a low probability ($p < 0.05$) of random data having a higher IndVal score than the bootstrapped observed data. Community change points are also calculated after selecting (filtering) only the taxa that exceeded purity and reliability requirements for the increasing (fsum(z+)) and declining (fsum(z-)) change points, where f refers to filtered results. Narrow peaks around the maximum sum(z) or fsum(z) (filtered) scores indicate areas with synchronous change in individual taxa frequency and abundance and may indicate an ecological threshold along the environmental gradient being evaluated. More information on TITAN methods can be found in Baker and King (2013) or in the TITAN R package, TITAN2.

While TITAN is a powerful tool for community ecologists it requires additional analysis for comparing results from different regions, groups or treatments. Previously, researchers have used non-overlapping confidence intervals for change-points from TITAN output to indicate significant differences between groups (King *et al.* 2011). However, although non-overlapping confidence intervals can indicate significant differences, overlapping confidence intervals do not necessarily mean the null hypothesis should be accepted (Greenland *et al.* 2016; Schenker and Gentleman 2001).

We developed a new R package, pTITAN2, as an extension of the existing TITAN2 R package. The goal of pTITAN2 is to enable comparisons of TITAN output between treatments by permuting the observed data between treatments and rerunning TITAN on the permuted data. There are some limitations on the permutations, including (1) a sampling site cannot occur in a category more than once, the same limitation as in the original TITAN runs and (2) the original sample size distribution is maintained. This addresses potential sample size effects and enables comparisons between treatments with different sample sizes more accurately than using non-overlapping confidence intervals. A vignette is provided based on a dataset of macroinvertebrate data from California streams that fall along a gradient of watershed percent impervious cover. We compare change-points among different climate conditions (wet, average, and dry) based on the Palmer Drought Severity Index, which serve as the treatments in this example.

2. Methods

2.1 Operation

Like TITAN2, pTITAN2 was developed using the R programming language (RRID:SCR_001905). pTITAN2 has been tested on Windows, Mac OS, and Ubuntu on the latest R version (4.1.0 at time of writing) along with select old release (4.0 and 3.6) and development versions via github actions. It is recommended that users are familiar with the TITAN2 package operations before using pTITAN2.

The basic workflow for pTITAN2 is

1. Prepare and import the environmental gradient dataset into R
2. Prepare and import the taxonomic dataset into R

3. Preprocess raw taxonomic data to determine appropriate taxonomic level of resolution (occurrence function)
4. Select columns for the taxon level returned by occurrence function
5. Permute the data across treatment labels to generate list of lists
6. Set up cluster for parallel processing (optional)
7. Run TITAN2 series on original and permuted data sets
8. Analyze probability of exceeding observed difference in changepoint between treatments based on distribution of paired changepoint differences

2.2 Implementation

The first step of pTITAN2 is to provide the data about the environmental gradient in exactly the format as for TITAN, step 1). This can be either a single file or included in the taxonomic data file. Like TITAN2, taxonomic information should be provided as counts or density. Unlike TITAN2, pTITAN2 taxonomic data need to be provided as a code that is eight characters in length and captures four levels of hierarchical taxonomic classification information.

The pTITAN2 package provides four example data sets, two taxonomic and two environmental gradient (Table 1). These data sets are provided as raw csv files and as prepared R datasets.

You can gain access to the csv files via `system.file`

```
list.files(system.file("extdata", package = "pTITAN2"))
## [1] "C_IC_D_06_wID.csv" "C_IC_N_06_wID.csv" "CD_06_Mall_wID.csv"
## [4] "CN_06_Mall_wID.csv"
```

or get the data sets loaded into your environment via

```
data(C_IC_D_06_wID, C_IC_N_06_wID, CD_06_Mall_wID, CN_06_Mall_wID,
     package = "pTITAN2")

str(C_IC_D_06_wID) # Environmental Gradient, Dry Treatment
## 'data.frame':   251 obs. of  2 variables:
## $ StationID: chr  "308000" "707063" "707086" "500762" ...
## $ ImpCover : num  0.0151 0.0907 0.076 3.9944 2.5167 ...
str(C_IC_N_06_wID) # Environmental Gradient, Normal Treatment
## 'data.frame':   124 obs. of  2 variables:
## $ StationID: chr  "707059" "707076" "707219" "500004" ...
## $ ImpCover : num  0.05959 0.00695 8.46983 35.8094 12.97293 ...
dim(CD_06_Mall_wID) # Taxonomic, Dry Treatment
## [1] 251 501
dim(CN_06_Mall_wID) # Taxonomic, Normal Treatment
## [1] 124 501
```

The `CN_06_Mall.csv` (Chaparral Region, Treatment = Normal) file contains raw macroinvertebrate density data for 500 possible macroinvertebrate codes for each taxonomic level (class, order, family, genus). The `occurrences` function selects the codes that should be used for the `TITAN2::titan` run. The goal is to select the macroinvertebrate code with the most taxonomic detail having at least `n` occurrences. Only one macroinvertebrate code will be associated with the macroinvertebrate counts. For example, if there are at least five occurrences at the genus level, the family, order, and class codes would not be used in the `TITAN2::titan` run.

Table 1. Example data sets provided in pTITAN2.

R Data	csv File	Data Type	Region	Treatment
C_IC_D_06_wID	C_IC_D_06_wID.csv	Environmental Gradient	Chaparral	Dry
C_IC_N_06_wID	C_IC_N_06_wID.csv	Environmental Gradient	Chaparral	Normal
CD_06_Mall_wID	CD_06_Mall_wID.csv	Taxonomic	Chaparral	Dry
CN_06_Mall_wID	CN_06_Mall_wID.csv	Taxonomic	Chaparral	Normal

The names within the data set are expected to have the following structure:

- 8 characters in length
- characters 1 and 2 denote the class
- characters 3 and 4 denote the order
- characters 5 and 6 denote the family
- characters 7 and 8 denote the genus.

If no information at a level exists, use “00” to hold the place. For example: A code that is ‘Bi000000’ is the Bivalvia class, while BiVe0000 is the Bivalvia class, Veneroida order. BiVeSh00 is the Bivalvia class, Veneroida order, Sphaeriidae family. BiVeSh01 is a genus within that family.

The first new function provided by pTITAN2 is `occurrences`. Taking the taxonomic data as an input, the return of `occurrences` is a `data.frame` with the taxon, the class, order, family, and genus split out into individual columns, and the count of occurrences within the provided taxonomic data set. `TITAN2::titan` recommends all taxonomic groups have at least five observations (Baker and King 2010). Thus, `occurrences` returns only taxa with at least `n` observations (where `n` defaults to five). The taxonomic code chosen for analysis should be at the finest possible resolution. For example, if a macroinvertebrate count has at least five occurrences in a genus code, the family, order, and class codes associated with these counts should be removed. Further, if there are too few counts at the genus level, but at least five counts at the family level- the family code would be retained, and the order and class codes would be removed.

The second new function provided by pTITAN2 is the `permute` function which provides a list of permuted sets of taxa and environmental gradients. This function is used with categorical environmental variables (treatments), such as Wet/Dry or Urban/Rural. The function permutes the treatment labels across the data such that each station has a non-zero probability of being assigned to each treatment, and the stations are unique within each treatment and replication. There are some limitations on the permutations generated by `permute`. First, a site cannot occur in a category more than once within a permutation. Second, the original sample size distribution is maintained. These limitations address potential sample size effects in TITAN, where treatments with low sample sizes have wide confidence intervals and variable change points compared to treatments with high sample sizes, and enable comparisons between treatments with different sample sizes.

For example, assume we have sites A, B, C, D, and E with treatments 1 and 2 (Table 2). Let `Trt0` denote the initial treatment labels for the sites and `Trt1`, ..., `Trt4` denote permuted treatment labels. For sites A and C, each permuted set of treatment labels consist of one row for label 1 and one row for label 2. For sites B, D, and E, the initial observations were for treatments 1, 2, and 2 respectively. The balance of these labels is maintained across the permutations.

After permutations, clusters can be used for parallel processing of `TITAN::titan()` calls. This can be advantageous as `TITAN::titan()` calls can be time and computationally expensive. Following the needed `TITAN::titan()` calls the differences between treatment change points in the observed data can be compared to the differences between treatment change points in the permuted stat to determine if the observed treatment differences are statistically significant.

Table 2. Example distribution of sites and permuted treatment labels. Trt = treatment.

site	trt0	trt1	trt2	trt3	trt4
A	1	1	2	1	1
A	2	2	1	2	2
B	1	2	2	2	1
C	1	1	2	2	1
C	2	2	1	1	2
D	2	2	2	1	2
E	2	1	1	2	2

3. Example

Here we present an example showing implementation of pTITAN2. We will describe the provided example data sets and how to use the `occurrences()` and `permute()` functions.

To reproduce the examples in this vignette you will need to load and attach the pTITAN2 and magrittr namespaces. Other namespaces are used explicitly, loaded (not attached) here.

```
library(pTITAN2)
library(magrittr)
loadNamespace("data.table")
## <environment: namespace:data.table>
loadNamespace("dplyr")
## <environment: namespace:dplyr>
loadNamespace("tidyr")
## <environment: namespace:tidyr>
```

3.1 Example data

Example data provided within the pTITAN2 package were based on publicly available stream macroinvertebrate data from California. The data include existing macroinvertebrate abundances from the California Environmental Data Exchange Network (CEDEN, last accessed 30 June 2017), and the Southern California Coastal Water Research Project (SCCWRP) (Fetscher *et al.* 2014). Samples in the CEDEN dataset were collected between 2000 and 2016, and samples from the SCCWRP dataset were collected between 1997 and 2011. Both data sets were generated using probabilistic sampling designs and are expected to be representative of streams in the region (Figary *et al.* 2021).

For this example, data were extracted for California's Chaparral Region (Ode *et al.* 2011). Sample observations were divided into one of three classes based on the precipitation regime for the sampling year using the Palmer Drought Severity Index (PDSI). The PDSI was determined for each sampling event using monthly PDSI data from the National Oceanic and Atmospheric Administration (NOAA, last accessed 21 December 2016) (RRID:SCR_009427) and climate divisions from the National Climatic Data Center (USGS 2004, last accessed 21 December 2016). We classified all sampling events as dry if PDSI was less than -2, normal for PDSI between -2 and 2, or wet if PDSI was greater than 2 and these classifications were used as treatments for the permutations. These cutoffs correspond to NOAA categories of moderate drought and unusually wet soil, respectively. The environmental gradient of interest was percent impervious cover in the upstream watershed, in this case defined by the National Land Cover Datasets (NLCD, Homer *et al.* (2007, 2015), with values interpolated between NLCD years of record (2001, 2006, 2011). Impervious area additions beyond 2011 were estimated as 50% of disturbed area for construction sites as documented in the California Stormwater Multiple Application and Report Tracking System (SMARTS dataset, CalEPA).

For running pTITAN2, the example data sets have a separate csv or pre-built R data sets (Table 1), for the environmental variable, in this case percent impervious cover, and macroinvertebrate density data. The data structure that is shown here is not required for pTITAN2 and instead the environmental variables and treatments could be in a single data file and subdivided as desired. Separate data files are provided for each 'treatment' that is explored including data from either drought (dry) or normal precipitation years in the Chaparral region of California.

3.2 Function occurrences

The taxonomic sets, `CD_06_Mall_wID` and `CN_06_Mall_wID`, contains raw macroinvertebrate density data for 500 possible macroinvertebrate codes for each taxonomic level (class, order, family, genus).

```
dim(CD_06_Mall_wID)
## [1] 251 501

# top 4 rows and first 10 columns
CD_06_Mall_wID[1:4, 1:10]
##   StationID Ar000000 Bi000000 BiUn0000 BiUnUi00 BiUnUi01 BiVe0000 BiVeCa00
## 1 308000 48.484848      0      0      0      0      0      0
## 2 707063 97.979798      0      0      0      0      0      0
## 3 707086 1.010101      0      0      0      0      0      0
## 4 500762 8.080808      0      0      0      0      0      0
##   BiVeCa01 BiVeSh00
## 1      0      0
## 2      0      0
## 3      0      0
## 4      0      0
```

The `occurrences` function selects the codes that should be used for the `TITAN2::titan` run. The goal is to select the macroinvertebrate code with the most taxonomic detail having at least `n` occurrences. Only one macroinvertebrate code will be associated with the macroinvertebrate counts. For example, if there are at least five occurrences at the genus level, the family, order, and class codes would not be used in the `TITAN2::titan` run.

The data are parsed within the `occurrences` call and return a data.frame with each taxon code split into its components and the frequency of the taxon within the data set. This is an extension for deciding the taxonomic detail to be included in a TITAN run based on `minSplt` in TITAN. `minSplt` is minimum number of occurrences that TITAN is looking for taxa to have across the provided sites. The `minSplt` default in TITAN is five and, as noted by [Baker and King \(2010\)](#), should never drop below three. The default for occurrences is `minSplt = n = 5`.

```
head(occurrences(CN_06_Mall_wID[, -1]))
##      taxon Class Order Family Genus count
## 1 Ar000000    Ar    00    00    00   115
## 2 BiVeCa01    Bi    Ve    Ca    01    16
## 3 BiVeSh00    Bi    Ve    Sh    00    42
## 4 EnHoTt01    En    Ho    Tt    01    19
## 5 GaBaAc01    Ga    Ba    Ac    01    14
## 6 GaBaLy02    Ga    Ba    Ly    02     5
```

Compare these results to working with the raw data. For example purposes we present the summary of the raw data twice, once using tidyverse syntax and once using data.table syntax.

```
# Tidyverse
CN_06_Mall_wID %>%
  dplyr::select(-StationID) %>%
  tidyr::pivot_longer(cols = dplyr::everything(), names_to = 'taxon', values_to = 'count') %>%
  dplyr::mutate(
    Class = substr(.data$taxon, 1L, 2L),
    Order = substr(.data$taxon, 3L, 4L),
    Family = substr(.data$taxon, 5L, 6L),
    Genus = substr(.data$taxon, 7L, 8L)
  ) %>%
  dplyr::group_by(.data$Class, .data$Order, .data$Family, .data$Genus) %>%
  dplyr::summarise(
    taxon = unique(.data$taxon),
    count = sum(.data$count > 0),
    .groups = "keep"
  ) %>%
  dplyr::ungroup() %>%
  dplyr::arrange(.data$Class, .data$Order, .data$Family, .data$Genus)
## # A tibble: 500 x 6
##   Class Order Family Genus taxon      count
##   <chr> <chr> <chr> <chr> <chr> <int>
## 1 Ar    00    00    00    Ar000000    115
## 2 Bi    00    00    00    Bi000000     46
## 3 Bi    Un    00    00    BiUn0000     0
## 4 Bi    Un    Ui    00    BiUnUi00     0
## 5 Bi    Un    Ui    01    BiUnUi01     0
## 6 Bi    Ve    00    00    BiVe0000     46
## 7 Bi    Ve    Ca    00    BiVeCa00     16
## 8 Bi    Ve    Ca    01    BiVeCa01     16
## 9 Bi    Ve    Sh    00    BiVeSh00     42
## 10 Br   00    00    00    Br000000     0
## # ... with 490 more rows

# data.table
taxon_count <- data.table::copy(CN_06_Mall_wID)
data.table::setDT(taxon_count)
data.table::set(taxon_count, j = "StationID", value = NULL)
for(j in as.integer(which(sapply(taxon_count, is.integer)))) {
  data.table::set(taxon_count, j = j, value = as.numeric(taxon_count[[j]]))
}

taxon_count <- data.table::melt(taxon_count, variable.factor = FALSE, measure.vars = names(taxon_count),
  variable.name = "taxon")
taxon_count[, value := sum(value > 0), keyby = .(taxon)]
taxon_count <- unique(taxon_count)

data.table::set(taxon_count, j = "Class", value = substr(taxon_count[["taxon"]], 1L, 2L))
data.table::set(taxon_count, j = "Order", value = substr(taxon_count[["taxon"]], 3L, 4L))
data.table::set(taxon_count, j = "Family", value = substr(taxon_count[["taxon"]], 5L, 6L))
data.table::set(taxon_count, j = "Genus", value = substr(taxon_count[["taxon"]], 7L, 8L))

data.table::setkeyv(taxon_count, c("taxon", "Class", "Order", "Family", "Genus"))

taxon_count[grepl("^(Ar|Bi).+", taxon)]
##      taxon value Class Order Family Genus
## 1: Ar000000   115    Ar    00    00    00
## 2: Bi000000    46    Bi    00    00    00
## 3: BiUn0000     0    Bi    Un    00    00
## 4: BiUnUi00     0    Bi    Un    Ui    00
## 5: BiUnUi01     0    Bi    Un    Ui    01
## 6: BiVe0000    46    Bi    Ve    00    00
## 7: BiVeCa00    16    Bi    Ve    Ca    00
## 8: BiVeCa01    16    Bi    Ve    Ca    01
## 9: BiVeSh00    42    Bi    Ve    Sh    00
```


Note that for the Ar class there is only one row with no order, family, or genus level information. Compare to the Bi class where the Un order has no presence counts and is thus not reported in object returned from `occurrences`. BiVeCa01 has counts and will be reported but BiVeCa00 should not be reported. BiVe0000 and Bi000000 should not be reported as `occurrences` as preference for the codes with family and genus level information.

3.3 Function `permute`

The function `permute` is used to generate a list of permuted sets of taxa and environmental gradients. Function parameters include a list of data frames containing taxa for each treatment group, a list of data frames containing the associated environmental gradient variables, and the site ids. Before we can run `permute`, we need to import the environmental gradients data.

```
eg <-
  permute(taxa = list(CD_06_Mall_wID, CN_06_Mall_wID),
          envs = list(C_IC_D_06_wID, C_IC_N_06_wID),
          sid = "StationID")
str(eg, max.level = 2)
## List of 2
## $ Treatment1:List of 2
## ..$ env :'data.frame': 251 obs. of 1 variable:
## ..$ taxa:'data.frame': 251 obs. of 500 variables:
## $ Treatment2:List of 2
## ..$ env :'data.frame': 124 obs. of 1 variable:
## ..$ taxa:'data.frame': 124 obs. of 500 variables:
## - attr(*, "minTaxonFreq")= num [1:2] 0 0
```

The return of `permute` is a list of lists. The first level denotes the treatment; in this example Treatment1 is “dry” and Treatment2 is “normal” – the order of the input data sets. The second level contains the data.frames with environmental and taxonomic data.

3.4 Running `TITAN2::titan`

The most computationally expensive part of this work is calling `TITAN2::titan` many, many times. A good option is to use the parallel package to send the task of permuting the data and running `TITAN2::titan()` to individual processing cores. That is system dependent and left to the end user to implement. For an example of generating the permutations with `TITAN2::titan()` see the example script provided at:

```
system.file("example-scripts/permutation_example.R", package = "pTITAN2")
## [1] "/Users/pdewitt/Library/R/x86_64/4.1/library/pTITAN2/example-scripts/permutation_example.R"
"
```

That file will generate the provided data set `permutation_example` with ten rows from ten permutations of the example data set.

```
permutation_example
##      trtlcpsumz- trtlcpsumz+ trtlcount trt2cpsumz- trt2cpsumz+ trt2count
## 1  0.24450828  0.9322147      251  0.21052157   7.8379526      124
## 2  0.20733907  0.9322147      251  0.20307915   0.6447989      124
## 3  0.17739530  0.6017841      251  0.20532745   1.9463146      124
## 4  0.43918849  0.9195990      251  0.20949813   4.2981178      124
## 5  0.20733907  1.0749101      251  0.17238643   0.4298088      124
## 6  0.20941096  0.6012265      251  0.20675722   1.1443447      124
## 7  0.06755573  0.6014938      251  0.29043078   2.1666044      124
## 8  0.14227803  0.6222103      251  0.26105344   0.3336509      124
## 9  0.22585229  0.6017841      251  0.07078791  10.5727762      124
## 10 0.24486085  0.6017841      251  0.21857924   2.0596616      124
##      permutation
## 1             1
## 2             2
## 3             3
## 4             4
## 5             5
## 6             6
## 7             7
## 8             8
## 9             9
## 10            10
```

The results are the increasing and decreasing taxa sumz values. In this example only ten permutations are used, and TITAN bootstrapping is limited to five iterations. In an actual analysis these values should be much higher. This process is very computationally intensive and can take hours or days to run depending on the available computing power and the number of bootstraps and permutations used.

If you have three or more treatments and need to permute over them with the condition that no station will be in the same treatment more than once on any particular permutation and that all treatment labels are viable for each station then you can still use the `permute` function.

3.5 Analyzing the results

The output from the code in section 3.4 can then be used to compare the differences in change-point values for treatments from the observed samples versus the permuted samples. A p-value test can be run on these data to test for statistically significant differences between the treatment effects.

```
# Run TITAN on the observed data
# limited to 2 cores for CRAN policies, end users can use more cores
# the following uses the tidyverse dialect, load and attach the magrittr
# namespace, use dplyr and tidyr namespaces explicitly.
library(magrittr)

CD_obs <-
  TITAN2::titan(
    env      = C_IC_D_06_wID[["ImpCover"]] # chaparral envgrad dry
    , txa     = subset(CD_06_Mall_wID, select = occurrences(CD_06_Mall_wID[, -1], n =
6L)$taxon)
    , ncpus   = 2
    , numPerm = 50
    , nBoot   = 50
  )
## Screening taxa...
## taxa frequency screen complete.
## Partitioning along gradient...
## Calculating observed IndVal maxima and class values...
## Calculating IndVals using mean relative abundance...
## Permuting IndVal scores...
## Summarizing observed results...
## Estimating taxa change points using z-score maxima...
## Bootstrap resampling in parallel using 2 CPUs... no index will be printed to screen.
## Proportion of pure and reliable taxa = 0.5732

CN_obs <-
  TITAN2::titan(
    env      = C_IC_N_06_wID[["ImpCover"]] # chaparral envgrad dry
    , txa     = subset(CN_06_Mall_wID, select = occurrences(CN_06_Mall_wID[, -1], n =
6L)$taxon)
    , ncpus   = 2
    , numPerm = 50
    , nBoot   = 50
  )
## Screening taxa...
## taxa frequency screen complete.
## Partitioning along gradient...
## Calculating observed IndVal maxima and class values...
## Calculating IndVals using mean relative abundance...
## Permuting IndVal scores...
## Summarizing observed results...
## Estimating taxa change points using z-score maxima...
## Bootstrap resampling in parallel using 2 CPUs... no index will be printed to screen.
## Proportion of pure and reliable taxa = 0.1129

# create a table of the median change point from TITAN for calculating the p-values
TITAN_med <-
  rbind(
    cbind(as.data.frame(CD_obs$sumz), "run" = "Tr1_CD")
    , cbind(as.data.frame(CN_obs$sumz), "run" = "Tr2_CN")
  )
TITAN_med[["sumz"]] <- sub("(\\d)$", "\\1", rownames(TITAN_med))

TITAN_med <-
  TITAN_med %>%
  dplyr::select(run, sumz, `0.50`) %>%
  tidyr::pivot_wider(names_from = run, values_from = "0.50") %>%
  dplyr::mutate(T1T2_abs = abs(Tr1_CD - Tr2_CN)) %>%
  print
```

```
## # A tibble: 4 × 4
##   sumz   Tr1_CD Tr2_CN T1T2_abs
##   <chr>   <dbl> <dbl>   <dbl>
## 1 sumz-   0.259 0.0851   0.174
## 2 sumz+   0.782 3.74     2.96
## 3 fsumz-   0.280 0.0605   0.219
## 4 fsumz+   0.602 2.59     1.98

# Create a summary table of the permutation data
tr1_filt <-
  permutation_example %>%
  dplyr::select(permutation, `trt1cpsumz-`, `trt1cpsumz+`) %>%
  tidyr::pivot_longer(cols = `trt1cpsumz-`:`trt1cpsumz+`,
                      values_to = "Tr1_CD",
                      names_to = "sumz")

tr1_filt[which(tr1_filt$sumz == "trt1cpsumz-"), 2] <- "fsumz-"
tr1_filt[which(tr1_filt$sumz == "trt1cpsumz+"), 2] <- "fsumz+"

tr2_filt <-
  permutation_example %>%
  dplyr::select(permutation, `trt2cpsumz-`, `trt2cpsumz+`) %>%
  tidyr::pivot_longer(cols = `trt2cpsumz-`:`trt2cpsumz+`,
                      values_to = "Tr2_CN",
                      names_to = "sumz")

tr2_filt[which(tr2_filt$sumz == "trt2cpsumz-"), 2] <- "fsumz-"
tr2_filt[which(tr2_filt$sumz == "trt2cpsumz+"), 2] <- "fsumz+"

# mutate the data and create a summary table for calculating p-values
out_perm <-
  dplyr::left_join(tr1_filt, tr2_filt) %>%
  dplyr::mutate(T1T2_abs = abs(Tr1_CD - Tr2_CN)) %>%
  dplyr::select(-permutation)
## Joining, by = c("permutation", "sumz")

# Calculate the p-values
dplyr::tibble(treatment = "T1T2_CDCN",
             "fsumz-" = (sum(
               dplyr::filter(out_perm, sumz == "fsumz-")$T1T2_abs >
               (dplyr::filter(TITAN_med, sumz == "fsumz-")$T1T2_abs)
             ) + 1) / 11, "fsumz+" =
             (sum(
               dplyr::filter(out_perm, sumz == "fsumz+")$T1T2_abs >
               (dplyr::filter(TITAN_med, sumz == "fsumz+")$T1T2_abs)
             ) + 1) / 11)
## # A tibble: 1 × 3
##   treatment `fsumz-` `fsumz+`
##   <chr>     <dbl>   <dbl>
## 1 T1T2_CDCN 0.26300 0.37400
```

4. Conclusions

TITAN is used in ecological studies to determine individual taxa and community level change points across an environmental gradient for both taxa that increase with the increasing environmental gradient and taxa that decrease with the increasing environmental gradient (Baker and King 2010). pTITAN2 was developed as an extension of TITAN to enable comparing TITAN results between different treatments, including those with variable sample sizes, by permuting the observed data between the treatments and then rerunning TITAN on the permuted dataset. This allows for statistically determining difference between the treatments without using overlapping confidence intervals, which can be problematic and can lead to accepting the null hypothesis more frequently than statistically necessary.

Data availability

Underlying data

Zenodo: pTITAN2. <https://doi.org/10.5281/zenodo.5894746> (Figary *et al.* 2021)

This project contains the following underlying data:

- CD_06_Mall_wID.csv (stream macroinvertebrate data from the Chapparral region of California collected during dry years)
- CN_06_Mall_wID.csv (stream macroinvertebrate data from the Chapparral region of California collected during normal precipitation years)

- C_IC_D_06_wID.csv (environmental gradient data (watershed percent imperviousness) for sites in CD_06_Mall_wID.csv)
- C_IC_N_06_wID.csv (environmental gradient data (watershed percent imperviousness) for sites in CN_06_Mall_wID.csv)

Extended data

Zenodo: pTITAN2. <https://doi.org/10.5281/zenodo.5894746> (Figary *et al.* 2021)

This project contains the following extended data:

- permutation_example.R (Results from a permutation example)

Data are available under the terms of the [Creative Commons Attribution 4.0 International license \(CC-BY 4.0\)](#).

Software availability

Source code available from: <https://github.com/USEPA/pTITAN2>

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.5894746>

License: [Creative Commons Attribution 4.0 International](#)

Author contributions

- Stephanie Figary: Data curation, Methodology, Software, Formal Analysis, Writing
- Naomi Detenbeck: Conceptualization, Funding Acquisition, Writing, Supervision
- Peter DeWitt: Software development, Writing

Acknowledgments

This is contribution number ORD-041368 of the Atlantic Coastal Environmental Sciences Division, Center for Environmental Measurement and Modeling, Office of Research and Development, U.S. Environmental Protection Agency. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

References

- Baker ME, King RS: **A New Method for Detecting and Interpreting Biodiversity and Ecological Community Thresholds.** *Methods Ecol. Evol.* 2010; **1**(1): 25–37.
[Publisher Full Text](#)
- Baker ME, King RS: **Of TITAN and Straw Men: An Appeal for Greater Understanding of Community Data.** *Freshwater Science.* 2013; **32**(2): 489–506.
[Publisher Full Text](#)
- Dufrène M, Legendre P: **Species Assemblages and Indicator Species: The Need for a Flexible Asymmetrical Approach.** *Ecol. Monogr.* 1997; **67**(3): 345–366.
[Publisher Full Text](#)
- Fetscher AE, Sutula M, Sengupta A, *et al.*: *Linking Nutrients to Alterations in Aquatic Life in California Wadeable Streams.* Washington, DC: US Environmental Protection Agency, Office of Research; Development; 2014. EPA/600/R-14/043.
- Figary S, DeWitt P, Detenbeck N: **pTITAN2 (Version 1).** Zenodo. 2021.
[Publisher Full Text](#)
- Gauch HG, Gauch HG: *Multivariate Analysis in Community Ecology.* United Kingdom: Cambridge University Press; 1982.
- Greenland S, Senn SJ, Rothman KJ, *et al.*: **Statistical Tests, p Values, Confidence Intervals, and Power: A Guide to Misinterpretations.** *Eur. J. Epidemiol.* 2016; **31**(4): 337–350.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Homer C, Dewitz J, Fry J, *et al.*: **Completion of the 2001 National Land Cover Database for the Conterminous United States.** *Photogramm. Eng. Remote. Sens.* 2007; **73**(4): 337.
- Homer C, Dewitz J, Yang L, *et al.*: **Completion of the 2011 National Land Cover Database for the conterminous United States – Representing a decade of land cover change information.** *Photogramm. Eng. Remote. Sens.* 2015; **81**(0): 345–354.
- King RS, Baker ME, Kazyak PF, *et al.*: **How Novel Is Too Novel? Stream Community Thresholds at Exceptionally Low Levels of Catchment Urbanization.** *Ecol. Appl.* 2011; **21**(5): 1659–1678.
[PubMed Abstract](#) | [Publisher Full Text](#)
- King RS, Richardson CJ: **Integrating Bioassessment and Ecological Risk Assessment: An Approach to Developing Numerical Water-Quality Criteria.** *Environ. Manag.* 2003; **31**(6): 795–809.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Ode PR, Kincaid TM, Fleming T, *et al.*: *Ecological Condition Assessments of California's Perennial Wadeable Streams, Highlights from the Surface Water Ambient Monitoring Program's Perennial Streams Assessment (PSA) (2000–2007).* A Collaboration Between the State Water Resources Control Board's Non-Point Source Pollution Control Program (NPS Program), Surface Water Ambient Monitoring Program (SWAMP), California Department of

Fish and Game Aquatic Bioassessment Laboratory, and the US Environmental Protection Agency; 2011.

Qian SS, King RS, Richardson CJ: **Two Statistical Methods for the Detection of Environmental Thresholds**. *Ecol. Model.* 2003; **166**(1-2): 87-97.

[Publisher Full Text](#)

Schenker N, Gentleman JF: **On Judging the Significance of Differences by Examining the Overlap Between Confidence Intervals**. *Am. Stat.* 2001; **55**(3): 182-186.

[Publisher Full Text](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 01 June 2022

<https://doi.org/10.5256/f1000research.88608.r137758>

© 2022 Braak C. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cajo J. F. ter Braak

Biometris, Wageningen University & Research, Wageningen, The Netherlands

General:

TITAN is an R software library that allows ecologists to obtain taxon-specific and a community-level threshold (i.e. point of maximum change along an environmental gradient, with change defined on the basis of indicator values (INDVAL) of Dufrene & Legendre or its maximum across taxa). TITAN provides both estimates and confidence intervals based on bootstraps. The current paper attempts to describe an add-on to TITAN called pTITAN, that aims to judge the statistical significance of differences in thresholds between two (or perhaps more?) regions. The need for such an add-on is well motivated as overlap or 'no overlap' of confidence intervals does not fully determine statistical significance.

The pTITAN R library contains two functions and a complicated and long example script with example data. As a reviewer I was interested in getting a p-value for the example data using the example scripts, but that was not easy; the example script on CRAN could not load the data and the example code in the paper could not be copied. Moreover, the example did not run as they needed a Unix-like computer OS and I use a Windows PC. In conclusion about the software: a wrapper function that runs on both Mac and Windows is needed so that the example script can be very short.

The issue that I have with the paper is exemplified in this sentence in section 3 titled Example: "Here we present an example showing implementation of pTITAN2." The example should show the usage of pTITAN2 and the implementation should have been described already, however there is none, there are only two basic functions and a script. The scripts looks like pTITAN to me.

I was also asking myself the question what I expect from a paper describing software. In my view, it should succinctly but fully describe the methods used and their implementation and provide an example. The current version of the paper seems to me to replicate the vignette of pTITAN, without giving a full description of the method; I am still uncertain about what is permuted (unrestricted or within bins of the environmental gradient?) and what role the bootstrap in TITAN

has in pTITAN. A simple mention/reference/link in the TITAN package to pTITAN would suffice. Simultaneously, one should work on the help files of both TITAN and pTITAN. For example, the occurrences function in pTITAN has argument data with description "A data.frame wit" "(i.e. incomplete, unfinished work) and TITAN has function titan with an argument txa with description "txa A site by taxon matrix containing observed counts at each sampling location." The latter leaves the question what is a site and what is sampling location; by now, I understand that they are identical. Also the default value of ivTOT is unclear in the help. Note that there is a permute package in R that could be used if the data sets are combined.

I had no idea about the type of data that was needed for pTITAN. It needs an environmental gradient and a treatment, it seems. A pictorial summary of the data would help, with treatment simply named 'region' as this seems the most likely application. What about paired data (a before and after situation in the same region)? Please be more explicit.

I wondered about the issue that the distribution of the environmental variable x in the one region might differ somewhat from that distribution in the other region (so that there is correlation between region and x. In such case, simple permutation of region labels might yield an invalid test; see (Fieberg et al. 2020¹; ter Braak 2021²).

TITAN discretizes the environmental variable based on a minimum number of sites (minSplt). I wondered how that works with two regions, each having/giving different discretization's.

A permutation test needs a test statistic. I missed explicit mention of the test statistics in the paper.

I thought the paper was on differences in threshold values but I see more on differences in max z- and max z+ values.

It is unclear to me the rationale in R to use a list of four data sets as argument, one data frame or one for txa and one for env, region should suffice.

Details:

F1000research: please provide also a version with line numbers!

"statistical differences": What are these? "determine statistical differences" -> "determine statistical significance of a difference "

"different regions, groups or treatments" Say explicitly what terms you will use later on. And give a pictorial overview of the data. Be explicit on any pairing of the data (paired data, unpaired).

"There are some limitations on the permutations..." This should be under Methods. And I do not understand this paragraph. How or why can point 2) be a limitation? Please explain fully how pTITAN does the permutation test.

"category" in the same paragraph as the above quotation. Category of what? Of the environmental

gradient or of the treatment groups?

Methods:

How cumbersome that the user needs to do all via a complex example scripts. I recommend automating.

Where in the script(s) was there an example of point 4 (selecting and perhaps summing to the right taxon level)?

Elaborate on point 5. I do not understand what "permute the data across treatment labels" means. You mean to permute/shuffle the treatment labels in the data. I also saw permutation of taxon data somewhere.

Point 8. What is the purpose of "paired" here. You mean the permutation distribution of change point differences?

Then, there is a clumsy section with R code to enter the data. I do not believe this is needed. Table 1 and the naming convention feels clumsy. One or two files with a column for env/x and region would suffice, i.e. no table needed.

Table 2:

What is the purpose of the table, except exemplifying what a shuffle of labels may result in? Example distribution in the caption: I do not see a distribution.

"Permutated"? I believe this should read "permuted".

Explain why sites A and C occur twice and the others only once. What is the intended design?

The occurrence function looks useful to me. I understand its utility without a need for "Compare these results to working with the raw data." and the following code examples. These can be considered for deletion.

"A p-value test can be run on these data", it sounds as applying a t-test or similar, but you mean to say that you can derive a p-value from the permutation results by counting.

I saw the world median in the comments. Where and why is the median used?

More philosophically, a confidence interval and a statistical test both need a definition of the "true" or population value of the parameter at stake. How is the community threshold defined in large sample ($n = 10^6$ for example)? How sensitive is its definition on minSplt?

References

1. Fieberg JR, Vitense K, Johnson DH: Resampling-based methods for biologists. *PeerJ*. 2020; **8**: e9089 [PubMed Abstract](#) | [Publisher Full Text](#)
2. ter Braak C: Predictor versus response permutation for significance testing in weighted regression and redundancy analysis. *Journal of Statistical Computation and Simulation*. 2021. 1-19 [Publisher Full Text](#)

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

No

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

No

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

No

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

No

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Statistical ecology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research