TeddyPi: Transposable Element detection and discovery for Phylogentic inference

TeddyPi is a modular pipeline to collect predicted transposable element (TE) and structural variation (SV) calls from a variety of programs and taxa. TeddiPi filters these variants for those of high-quality. Orthologous TE loci among a set of taxa are utilized to create a presence / absence matrix in NEXUS format for easy phylogenetic inference in phylogeny programs such as PAUP or SplitsTree.

By using the simple-but-powerful YAML syntax for configuration, TeddyPi can easily be adapted for other TE callers. Support for RetroSeq, Pindel and Breakdancer is included.

Dependencies

TeddyPi runs on Python 2.7. All Python dependencies can be installed using `pip install`. Version numbers TeddyPi has been tested with are given in parentheses.

- pybedtools (0.7.7)
- pyVCF (0.6.7)
- python-nexus (v 1.32)
- pyYAML (3.11)
- BioPython (1.66) http://biopython.org

Non Python dependency

- Bedtools (2.26) http://bedtools.readthedocs.io/en/latest/

Getting started

Installation

When dependencies are installed, TeddyPi can be obtained by cloning this git-repository:

```
git clone https://github.com/mobilegenome/teddypi
```

TeddyPi assumes that you have already called TEs and deletion from your NGS dataset. TeddyPi was developed for callsets generated by RetroSeq, Pindel and BreakDancer, but it can also be utilized for other programs. TeddyPi reads VCF (or pseudo-VCF) files. If your TE/SV caller does not outputs VCF files, you have to convert the output files. A conversion for breakdancer (based on breakdancer2vcf.py is included in the `utils/` directory

Run

To perform a full run, start the `tpi_wrapper.sh` in a BASH shell.

Input files

TeddyPi requires VCF files generated by TE/SV callers of your choices. If other file formats are generated by the TE/SV caller, it needs to be converted to VCF.

Also, TeddyPi requires BED files containing TE and other repeat annotation from the reference genome. These BED files are generated from the RepeatMasker output as described here. If you wish to filter for assembly gaps in the reference genome (which we highly recommend), also a BED file containing the gap coordinates in the reference genome is needed. Note, that all BED files should be sorted using `sort` or `bedtools sort`.

Modules

tpi_filter.py

Loads VCF files and applies filter operations as defined in YAML-formatted config file.

tpi_svintegration.py

This module loads filtered deletion calls files two SV callers and unifies it. Currently it is tailored towards Pindel and Breakdancer, but other SV can be implemented in the future. The module filters out regions with putative bad assembly quality.

tpi_ortho.py

The core functionality in of TeddyPi is to load a set of transposable element variants (TE) from VCF files. VCF files have to be created per sample, and are intersected by TeddyPi to create a presence / absence matrix.

tpi_unite.py

The module `tpi_unite.py` combines presence / absence matrices from Ref+ and Ref- calls to final matrix and generates a NEXUS file that can be utilized by phylogeny-software such as PAUP, SplitsTree4 or others.

tpi_helpers.py

Helper function used by TeddyPi.

utils

Several utilies are included to create VCF files from SV and TE callers that are often proprietary.

Configuration

TeddyPi is configured with configuration files in the YAML format (yaml.org). The main configuration is in `teddypi.yaml` stores the list of samples,list of programs and some parameters.

A detailed description on how to configure TeddyPi can be found in the wiki

```
```