PICTographPlus

1. Introduction

This is a tutorial for using PICTographPlus. The tool infers tumor clonal evolution from single or multi-region sequencing data by modeling the uncertainty of mutation cellular fraction (MCF) in small somatic mutations (SSMs) and copy number alterations (CNAs). Using a Bayesian hierarchical model, it assigns SSMs and CNAs to subclones, reconstructing tumor evolutionary trees that adhere to principles of lineage precedence, sum condition, and optional constraints based on sample presence. For deconvolution, PICTographPlus integrates tumor clonal tree structures with clone proportions across samples to resolve bulk gene expression data. It optimizes an objective function that minimizes discrepancies between observed and predicted sample-level gene expression while imposing a smoothness penalty, ensuring that closely related clones display greater gene expression similarity. Lastly, the tool conducts pathway enrichment analysis to identify statistically significant alterations in pathways connecting tumor clones.

---

2. Input data file format

PICTographPlus takes input data in multiple formats for flexible user inputs:

1) Three CSV files, one each for SSMs, CNAs, and germline heterozygous SNVs (RECOMMENDED)
2) Two csv files, one for SSM and one for CNA
3) A single csv file that contains SSM and CNA information.

2.1 Input data (DNA) for tumor evolution reconstruction

2.1.1 The recommended input for tumor evolution reconstruction is to provide individual files for SSMs, CNAs and germline heterozygous SNVs.

The SSM file should contain at least sample, mutation, total_reads, alt_reads, chrom, start, and end columns, with the purity column being optional.

```
head(read.csv(system.file('extdata/examples/example4_snv_with_purity.csv',
                          package = 'pictographPlus')))
#>    sample         mutation total_reads alt_reads chrom start  end purity
#> 1 sample1   chr1-57-clone1         100        67  chr1    57   57    0.8
#> 2 sample1  chr1-110-clone1         100        67  chr1   110  110    0.8
#> 3 sample1  chr1-167-clone1         100        67  chr1   167  167    0.8
#> 4 sample1  chr1-386-clone1         100        40  chr1   386  386    0.8
#> 5 sample1  chr1-441-clone1         100        40  chr1   441  441    0.8
#> 6 sample1 chr1-1276-clone1         100        40  chr1  1276 1276    0.8
```

The CNA file should contain sample, chrom, start, end, and tcn columns. The total copy number (tcn) can be inferred from many copy number callers. In case a copy number caller outputs the $log2$ ratio of a segment, the tcn can be calculate using $tcn = 2^{\log_2 R}$.

```
head(read.csv(system.file('extdata/examples/example4_cna.csv',
                          package = 'pictographPlus')))
#>    sample chrom start  end tcn
#> 1 sample1  chr1    1  200 3.6
#> 2 sample2  chr1    1  200 3.4
#> 3 sample3  chr1    1  200 3.4
#> 4 sample3  chr1  401 1500 1.4
#> 5 sample3  chr1 7001 7200 2.6
```

The SNV file contains the count of the heterozygous germline SNVs that has the information about the "chrom", the "position" of the germline heterozygous SNV, "ref" and "alt" allele, and the reference and altenative reads counts in germline (normal) sample as well as all other samples. Note: the sample name (sample1, sample2, … etc.) should matched the sample name used in the SSM and CNA file.

```
head(read.csv(system.file('extdata/examples/example4_SNP.csv',
                          package = 'pictographPlus')))
#>   chroms position ref alt germline_ref germline_alt sample1_ref sample1_alt sample2_ref sample2_alt
#> 1   chr1      178   A   C           50           50          72          28          71          29
#> 2   chr1      107   A   C           50           50          72          28          71          29
#> 3   chr1       84   A   C           50           50          28          72          29          71
#> 4   chr1       31   A   C           50           50          28          72          29          71
#> 5   chr1       18   A   C           50           50          72          28          71          29
#> 6   chr1      165   A   C           50           50          28          72          29          71
```

Obtaining Germline Heterozygous Positions The heterozygous germline positions can be obtained using tools such as **GATK HaplotypeCaller**. The reference and alternative read counts for each tumor sample can be retrieved using **samtools mpileup**.

We provide a Python script, `getPileUp.py`, to help generate the desired SNV file:

```
python getPileUp.py -v haplotype.vcf -b sample1.bam sample2.bam ... -o outputDir -f hg38.fa [--minreads
```

Parameter	Description	Option
-v	VCF output from HaplotypeCaller	**Required**
-b	Tumor BAM files for tumor samples	**Required**
-o	Output directory	**Required**
-f	Human reference genome	**Required**
–minreads	Minimum read count for both ref and alt to keep a site	Default: 3
–vaf	Minimum and maximum VAF for normal heterozygous sites	Default: 0.3 0.7

This script creates a folder `outputDir/pileup`, which contains a `germline_het.txt` file and all `pileup_summary.txt` files. These files are used to extract the germline heterozygous positions and convert them to the SNV file format described above. The resulting `germline_SNV.csv` can be used as input for **PICTographPlus**.

> **Note:** Ensure that the BAM file names (e.g., `sample1.bam`) match the sample names (e.g., `sample1`) used in the SSM and CNA files. If needed, rename the columns in `germline_SNV.csv` so they match the sample names in your SSM and CNA files.

2.1.2 Two csv files, one for SSM and one for CNA

The second option is to provide the SSM read counts and copy number alterations (CNA) in two separate files.

The SSM file should contain columns "sample", "mutation", "total_reads", "alt_reads", "chrom", "start", and "end". The "purity" column with be optional.

```
head(read.csv(system.file('extdata/examples/example3_snv_with_purity.csv',
                          package = 'pictographPlus')))
#>    sample         mutation total_reads alt_reads chrom start  end purity
#> 1 sample1   chr1-57-clone1         100        67  chr1    57   57    0.8
#> 2 sample1  chr1-110-clone1         100        67  chr1   110  110    0.8
#> 3 sample1  chr1-167-clone1         100        67  chr1   167  167    0.8
#> 4 sample1  chr1-386-clone1         100        40  chr1   386  386    0.8
#> 5 sample1  chr1-441-clone1         100        40  chr1   441  441    0.8
#> 6 sample1 chr1-1276-clone1         100        40  chr1  1276 1276    0.8
```

The CNA file should contain columns “sample”, “chrom”, “start”, “end”, “tcn” and “baf”.

```
head(read.csv(system.file('extdata/examples/example3_cna.csv',
                          package = 'pictographPlus')))
#>    sample chrom start  end tcn       baf
#> 1 sample1  chr1     1  200 3.6 0.2777778
#> 2 sample2  chr1     1  200 3.4 0.2941176
#> 3 sample3  chr1     1  200 3.4 0.2941176
#> 4 sample3  chr1   401 1500 1.4 0.2857143
#> 5 sample3  chr1  7001 7200 2.6 0.3846154
```

2.1.3 A single csv file that contains SSM and CNA information

The last option is to provide a single csv file that contains at least columns named “sample”, “mutation”, “total_reads”, “alt_reads”, “tumor_integer_copy_number”, and “cncf”. Set cncf to 0 if a mutation has no copy number alteration. Users can also provide an optional column “major_integer_copy_number” that provides the information of the integer copy number of the major allele. If “major_integer_copy_number” is not provided, it will be estimated using an internal function built in the package. Another optional column is “purity” column that provides the information of normal contamination of a sample.

NOTE: using this option will generate trees with SSMs only, CNA will not be assigned to clusters but only used for VAF correction.

```
head(read.csv(system.file('extdata/examples/example2_snv_with_purity.csv',
                          package = 'pictographPlus')))
#>    sample         mutation total_reads alt_reads tumor_integer_copy_number major_integer_copy_number
#> 1 sample1   chr1-57-clone1         100        67                         4                         3
#> 2 sample1  chr1-110-clone1         100        67                         4                         3
#> 3 sample1  chr1-167-clone1         100        67                         4                         3
#> 4 sample1  chr1-386-clone1         100        40                         2                         1
#> 5 sample1  chr1-441-clone1         100        40                         2                         1
#> 6 sample1 chr1-1276-clone1         100        40                         2                         1
```

2.2 Input data for bulk RNA expression

The RNA file should be a csv file of columns Gene, followed by the tumor samples (tumor sample name should match that of the genomic input), and lastly the read counts of a matched normal sample. The read counts should be normalized to transcript per million (TPM).

```
head(read.csv(system.file('extdata/examples/rna_example.csv',
                          package = 'pictographPlus')))
#>      Gene sample1 sample2 sampleN
#> 1    A1BG       0       5       0
#> 2    A1CF       0       0       0
#> 3     A2M      21      50       9
#> 4   A2ML1       0       0       0
#> 5 A3GALT2       0       0       0
#> 6  A4GALT       1       1       0
```

---

3. Run PICTographPlus

3.1 Running PICTographPlus in one step

PICTographPlus can be run using the function `runPICTographPlus`, which runs both tumor evolution reconstruction and clone-specific transcriptomic profile deconvolution. The required files include files for genomic data and RNA expression data.

```
runPICTographPlus(mutation_file, copy_number_file, SNV_file, rna_file, outputDir)
```

Detailed documentation for the function can be found:

```
help(runPICTographPlus)
```

3.2 Running PICTographPlus in multiple steps

Alternatively, the user can run tumor evolurion reconstruction and bulk RNA deconvolution in separate steps.

Tumor evolution reconstruction can be run using:

```
runPictograph(mutation_file, copy_number_file, SNV_file, outputDir)
```

Bulk RNA expression deconvolution can be run using:

```
runDeconvolution(rna_file, treeFile, proportionFile, purityFile, outputDir)
```

where the treeFile, proportionFile, and purityFile are outputs of `runPictograph` function. Users may also choose other tools to get these information. File formats can be found in section 6.

GSEA analysis using fgsea can be run using:

```
X_optimal <- read.csv(paste0(outputDir, "/clonal_expression.csv"),
                      row.names=1, check.names=FALSE)

runGSEA(X_optimal, outputDir, treeFile, GSEA_file)
```

where GSEA_file is a text file of pathways of interest that can be obtained from resources such as MSigDB.

---

4. Parameters

4.1 Parameters for runPICTographPlus and runPictograph

4.1.1 mutation_file [required]

a csv file that include information for SSMs. See section 2 for details.

4.1.2 copy_number_file

a csv file that include information for CNA. See section 2 for details.

4.1.3 SNV_file

a csv file that include information for germline heterozygous SNVs. See section 2 for details.

4.1.4 outputDir

output directory for saving all files.

4.1.5 sample_presence

whether or not to use sample presence to separate the mutations; default: TRUE

4.1.6 max_K

user defined maximum number of clusters; default: 10

4.1.7 alt_reads_thresh

minimum number of alternative read count for a SSM to be included in the analysis; default: 0

4.1.8 vaf_thresh

minimum VAF for a SSM to be included in the analysis; default: 0

4.1.9 smooth_cnv

whether or not to process copy number alterations across samples to unify the segment start and end postions; default: TRUE

4.1.10 autosome

to only include autosomes; default: TRUE

4.1.11 filter_cnv

whether or not to filter copy number alterations; default: TRUE

4.1.12 tcn_normal_range

range of total copy number considered as copy-neutral; default: c(1.75,2.3)

4.1.13 cnv_min_length

minimum length of copy number alterations for it to be included in analysis

4.1.14 purity_min

minimum purity for tumor samples; default: 0.2

4.1.15 min_mutation_per_cluster

minumum number of mutations in each cluster; default: 5

4.1.16 min_cluster_thresh

minimum MCF for each cluster; default: 0.05

4.1.17 cluster_diff_thresh

difference threshold to merge two clusters: default: 0.05

4.1.18 LOH

whether or not to include copy number segments that are copy neutral but LOH; default: FALSE

4.1.19 n.iter

number of iterations by JAGS; default: 5000

4.1.20 n.burn

number of burns by JAGS; default: 1000

4.1.21 thin

number of thin by JAGS; default: 10

4.1.22 inits

additional parameters by JAGS.

4.1.23 score

scoring function to estimate the number of clusters. silhouette or BIC; default: silhuette

4.1.24 mc.cores

number of cores to use for parallel computing; not applicable to windows; default: 8

4.1.25 driverFile

list of driver genes used for visualization.

4.1.26 cytobandFile

list of cytoband regions used for visualization.

4.2 Parameters for runPICTographPlus and runDeconvolution

4.2.1 rna_file [required]

bulk RNA file in integer read counts; rows are samples and columns are genes. See section 2 for details.

4.2.3 lambda

weights used in deconvolution step; default: 0.2

4.3 Parameters for runPICTographPlus and runGSEA

4.3.1 GSEA_file

geneset file in MSigDB .gmt format; the geneset name will show up in plotting

4.3.3 top_K

top_K significant pathways to be plotted as GSEA results; default: 5

4.3.3 n_permutations

number of permutations in fgsea; default: 10000

---

5. Output files

File name	Description
mcf.csv	estimated MCF for each cluster in each sample
clusterAssign.csv	cluster assignment of each SSM/CNA to each cluster
CN_results.csv	estimation of the integer and major copy number of CNAs
tree.csv	the tree with the highest score; all trees with tied highest score is available under all_trees directory.
subclone_proportion.csv	estimated proportion of each cluster in each sample
purity.csv	estimated purity for each sample, based on the tree structure
tree.png	the image of a tree with the best score
upsetR.png	the mutation profiles between samples; only available if number of samples is bigger than 1.
mcf.png	the MCF chain trace from JAGS
mutationClusterAssign.csv	table of all mutations information in all samples
clone_expression.csv	clone level gene expression for each clone
GSEA	directory contains all files from GSEA analysis

---

6. Additional file format

6.1 treeFile

A CSV file describing the tumor evolution tree used by the functions `runDeconvolution()` and `runGSEA()`.

edge	parent	child
root->1	root	1
1->2	1	2

6.2 proportionFile

A CSV file of subclone proportions used by `runDeconvolution()`. Each row corresponds to a clone, and each column to a sample.

	sample1	sample2
1	0.12	0.30
2	0.82	0.70

6.3 purityFile

A CSV file of tumor purity used by `runDeconvolution()`.

sample1	sample2
0.70	0.50

6.4 driverFile

A CSV file that contains driver mutation information used for plotting by `runPictograph()`.
If you use this file, ensure that the corresponding mutation file follows the format **gene_extraInformation**, where **gene** is the actual gene name. The `gene_type` column can be **oncogene** or **tumor_suppressor** (leave blank if unknown).

gene	chrom	start	end	gene_type
KRAS	chr12	25205246	25250936	oncogene
SMAD4	chr18	51028528	51085045	tumor_suppressor

6.5 cytobandFile

A TSV file of cytoband information used for plotting by `runPictograph()`.

chr	start	end	band	stain
chr1	0	2300000	p36.33	gneg
chr1	2300000	5300000	p36.32	gpos25