Below is one possible software specification outlining a set of Python API functions. Each function is described with its title, function name, a unique ID (or multiple IDs when one function serves several instructions), a purpose statement, a Python function header with a docstring (omitting the full implementation details), and a list of the instructions (by Workflow ID and Instruction ID) for which the function is used.

**Title of the Function 1: Aggregate Yield Statistics**  
**Function Name**: aggregate\_yield\_statistics  
**ID**: 4\_4, 5\_4, 15\_5, 20\_5  
**Purpose**:  
Aggregates wafer yield data based on specified grouping fields (such as probe configuration, test house, load board configuration, or tester–load board pairs) and computes summary statistics (e.g., mean, median, standard deviation).  
**Signature**:

def aggregate\_yield\_statistics(data: pd.DataFrame, group\_by: List[str], statistics: List[str] = ['mean', 'median', 'std']) -> pd.DataFrame:

"""

Aggregate yield data based on specified grouping fields and compute summary statistics.

Parameters:

data (pd.DataFrame): Input DataFrame containing yield information.

group\_by (List[str]): List of column names to group by (e.g., ['probe\_configuration'], ['test\_house'], ['load\_board'], or ['tester', 'load\_board']).

statistics (List[str]): List of statistical measures to compute (default: ['mean', 'median', 'std']).

Returns:

pd.DataFrame: A DataFrame with aggregated yield statistics.

"""

**Used For**:

* Workflow 4, Instruction 4
* Workflow 5, Instruction 4
* Workflow 15, Instruction 5
* Workflow 20, Instruction 5

**Title of the Function 2: Aggregate Logical Test Outcomes**  
**Function Name**: aggregate\_logical\_test\_outcomes  
**ID**: 14\_4  
**Purpose**:  
Aggregates test outcome data for logical tests within a specified time period (e.g., the current week) to compute the total numbers of passing and failing tests.  
**Signature**:

def aggregate\_logical\_test\_outcomes(data: pd.DataFrame, time\_period: str) -> pd.DataFrame:

"""

Aggregate logical test outcomes over a given time period.

Parameters:

data (pd.DataFrame): Input DataFrame containing logical test results.

time\_period (str): Time period over which to aggregate (e.g., 'current\_week').

Returns:

pd.DataFrame: A DataFrame with total counts of passing and failing logical tests.

"""

**Used For**:

* Workflow 14, Instruction 4

**Title of the Function 3: Aggregate Lot-Level Yield**  
**Function Name**: aggregate\_lot\_level\_yield  
**ID**: 39\_4, 63\_4, 19\_4  
**Purpose**:  
Aggregates wafer-level yield data for a given lot by summing or averaging the individual yields to provide an overall yield performance measure for that lot.  
**Signature**:

def aggregate\_lot\_level\_yield(data: pd.DataFrame, lot\_id: str, method: str = 'mean') -> float:

"""

Aggregate wafer yields for a specific lot to compute the overall yield performance.

Parameters:

data (pd.DataFrame): Input DataFrame with wafer yield information.

lot\_id (str): Identifier for the lot.

method (str): Aggregation method ('mean' or 'sum'); default is 'mean'.

Returns:

float: The overall yield performance for the specified lot.

"""

**Used For**:

* Workflow 39, Instruction 4
* Workflow 63, Instruction 4
* Workflow 19, Instruction 4

**Title of the Function 4: Aggregate and Rank Test Bins**  
**Function Name**: aggregate\_and\_rank\_test\_bins  
**ID**: 41\_5  
**Purpose**:  
Aggregates the number of failing dies per test bin for yield-failing lots and ranks the bins based on their contribution to overall failures.  
**Signature**:

def aggregate\_and\_rank\_test\_bins(data: pd.DataFrame, lot\_id: str) -> pd.DataFrame:

"""

Aggregate the failing die counts per test bin for a given yield-failing lot and rank the bins by failure contribution.

Parameters:

data (pd.DataFrame): Input DataFrame containing test bin failure data.

lot\_id (str): Identifier for the yield-failing lot.

Returns:

pd.DataFrame: A DataFrame with test bins ranked by their contribution to overall failures.

"""

**Used For**:

* Workflow 41, Instruction 5

**Title of the Function 5: Aggregate Failure Counts by Test Bin**  
**Function Name**: aggregate\_failure\_counts\_by\_bin  
**ID**: 52\_2, 38\_3  
**Purpose**:  
Groups failure records by test bin and counts the number of failures in each bin to help identify the most frequently failing test bins.  
**Signature**:

def aggregate\_failure\_counts\_by\_bin(data: pd.DataFrame, bin\_column: str) -> pd.DataFrame:

"""

Group the failure records by the specified test bin column and count the number of failures in each bin.

Parameters:

data (pd.DataFrame): Input DataFrame containing failure records.

bin\_column (str): The column name representing the test bin.

Returns:

pd.DataFrame: A DataFrame with each test bin and its corresponding failure count.

"""

**Used For**:

* Workflow 52, Instruction 2
* Workflow 38, Instruction 3

**Title of the Function 6: Aggregate Test Bin Counts for Lot**  
**Function Name**: aggregate\_test\_bin\_counts  
**ID**: 55\_5  
**Purpose**:  
Sums test bin counts across all wafers within a lot to produce an overall count of test bin occurrences for that lot.  
**Signature**:

def aggregate\_test\_bin\_counts(data: pd.DataFrame, lot\_id: str) -> int:

"""

Sum the test bin counts across all wafers for a given lot.

Parameters:

data (pd.DataFrame): Input DataFrame containing wafer-level test bin counts.

lot\_id (str): Identifier for the lot.

Returns:

int: The overall test bin count for the specified lot.

"""

**Used For**:

* Workflow 55, Instruction 5

**Title of the Function 7: Aggregate Failure Frequency Distribution by Test Bin**  
**Function Name**: aggregate\_failure\_frequency\_distribution  
**ID**: 71\_3  
**Purpose**:  
Counts the number of failing dies for each test bin, creating a frequency distribution to analyze the distribution of failures.  
**Signature**:

def aggregate\_failure\_frequency\_distribution(data: pd.DataFrame, bin\_column: str) -> pd.DataFrame:

"""

Count the number of failing dies for each test bin to create a frequency distribution.

Parameters:

data (pd.DataFrame): Input DataFrame containing failing die records.

bin\_column (str): The column name representing the test bin.

Returns:

pd.DataFrame: A DataFrame representing the frequency distribution of failures per test bin.

"""

**Used For**:

* Workflow 71, Instruction 3

**Title of the Function 8: Count Failure Occurrences by Test Measurement**  
**Function Name**: count\_failure\_occurrences\_by\_measurement  
**ID**: 76\_4  
**Purpose**:  
Counts the failure instances by grouping the data based on test measurement to determine which measurements most frequently lead to die failures.  
**Signature**:

def count\_failure\_occurrences\_by\_measurement(data: pd.DataFrame, measurement\_column: str) -> pd.DataFrame:

"""

Count the number of failure occurrences for each test measurement.

Parameters:

data (pd.DataFrame): Input DataFrame containing failure instance records.

measurement\_column (str): The column name representing the test measurement.

Returns:

pd.DataFrame: A DataFrame with each test measurement and its corresponding failure count.

"""

**Used For**:

* Workflow 76, Instruction 4

**Title of the Function 9: Quantify Failure Counts by Test Bin**  
**Function Name**: quantify\_failure\_counts\_by\_bin  
**ID**: 54\_4  
**Purpose**:  
Aggregates the number of failing dies per test bin to quantify which test bins are most significantly contributing to yield issues.  
**Signature**:

def quantify\_failure\_counts\_by\_bin(data: pd.DataFrame, bin\_column: str) -> pd.DataFrame:

"""

Quantify the failure counts by aggregating the number of failing dies for each test bin.

Parameters:

data (pd.DataFrame): Input DataFrame with failing die records.

bin\_column (str): The column name representing the test bin.

Returns:

pd.DataFrame: A DataFrame with each test bin and its corresponding quantified failure count.

"""

**Used For**:

* Workflow 54, Instruction 4

**Title of the Function 10: Summarize Dominant Failure Modes**  
**Function Name**: summarize\_dominant\_failure\_modes  
**ID**: 34\_8  
**Purpose**:  
Analyzes test bin data to identify and summarize the dominant failure modes (bins) in lots where yield performance drops.  
**Signature**:

def summarize\_dominant\_failure\_modes(data: pd.DataFrame, lot\_ids: List[str]) -> pd.DataFrame:

"""

Aggregate test bin data to identify the dominant failure modes in lots with yield drops.

Parameters:

data (pd.DataFrame): Input DataFrame containing test bin failure data.

lot\_ids (List[str]): A list of lot identifiers to analyze.

Returns:

pd.DataFrame: A DataFrame summarizing the dominant failure modes for the specified lots.

"""

**Used For**:

* Workflow 34, Instruction 8

**Title of the Function 11: Aggregate Failure Metrics by Test Program and Family**  
**Function Name**: aggregate\_failure\_metrics  
**ID**: 2\_6  
**Purpose**:  
Aggregates failure counts and computes statistics by grouping data according to test program and parametric test family for lots experiencing yield dips.  
**Signature**:

def aggregate\_failure\_metrics(data: pd.DataFrame, program\_column: str, test\_family\_column: str) -> pd.DataFrame:

"""

Summarize failure metrics by aggregating failure counts and statistics by test program and parametric test family.

Parameters:

data (pd.DataFrame): Input DataFrame containing failure records.

program\_column (str): Column name for the test program.

test\_family\_column (str): Column name for the parametric test family.

Returns:

pd.DataFrame: A DataFrame with aggregated failure metrics.

"""

**Used For**:

* Workflow 2, Instruction 6

**Title of the Function 12: Aggregate Failure Counts by Measurement Family**  
**Function Name**: aggregate\_failure\_counts\_by\_measurement\_family  
**ID**: 5\_5  
**Purpose**:  
For each test measurement family, counts the number of failing dies or failure events to quantify the frequency of failures when overall lot yield drops below 95%.  
**Signature**:

def aggregate\_failure\_counts\_by\_measurement\_family(data: pd.DataFrame, measurement\_family\_column: str) -> pd.DataFrame:

"""

Aggregate failure counts for each test measurement family.

Parameters:

data (pd.DataFrame): Input DataFrame containing failure records.

measurement\_family\_column (str): The column name representing the measurement family.

Returns:

pd.DataFrame: A DataFrame with each measurement family and its corresponding failure count.

"""

**Used For**:

* Workflow 5, Instruction 5

**Title of the Function 13: Aggregate Failure Counts by Test Program**  
**Function Name**: aggregate\_failure\_counts\_by\_program  
**ID**: 24\_4  
**Purpose**:  
Counts the total number of failing dies associated with each test program, thereby identifying the program contributing the most to die failures.  
**Signature**:

def aggregate\_failure\_counts\_by\_program(data: pd.DataFrame, program\_column: str) -> pd.DataFrame:

"""

Count the total number of failing dies for each test program.

Parameters:

data (pd.DataFrame): Input DataFrame containing failure records.

program\_column (str): The column name representing the test program.

Returns:

pd.DataFrame: A DataFrame with failure counts aggregated by test program.

"""

**Used For**:

* Workflow 24, Instruction 4

**Title of the Function 14: Aggregate Logical Test and Soft Bin Counts**  
**Function Name**: aggregate\_logical\_test\_and\_soft\_bin\_counts  
**ID**: 28\_5  
**Purpose**:  
Sums logical test failure counts and corresponding soft bin counts for each lot to help determine if logical tests are disproportionately contributing to soft bin counts on underperforming lots.  
**Signature**:

def aggregate\_logical\_test\_and\_soft\_bin\_counts(data: pd.DataFrame, lot\_id: str) -> pd.DataFrame:

"""

Sum up logical test failure counts and corresponding soft bin counts for a specified lot.

Parameters:

data (pd.DataFrame): Input DataFrame containing logical test and soft bin count data.

lot\_id (str): Identifier for the lot.

Returns:

pd.DataFrame: A DataFrame with the aggregated logical test and soft bin counts.

"""

**Used For**:

* Workflow 28, Instruction 5

**Title of the Function 15: Aggregate Test Bin Failure Frequency by Lot**  
**Function Name**: aggregate\_test\_bin\_failure\_frequency  
**ID**: 39\_6  
**Purpose**:  
For a given lot, aggregates failure data by test bin and calculates the failure frequency per bin, which is useful for tracking yield trends across multiple lots.  
**Signature**:

def aggregate\_test\_bin\_failure\_frequency(data: pd.DataFrame, lot\_id: str) -> pd.DataFrame:

"""

Aggregate failure data by test bin for a given lot and compute the failure frequency per bin.

Parameters:

data (pd.DataFrame): Input DataFrame containing failure records.

lot\_id (str): Identifier for the lot.

Returns:

pd.DataFrame: A DataFrame with each test bin and its computed failure frequency.

"""

**Used For**:

* Workflow 39, Instruction 6

This specification provides a modular design where each API function is tailored to handle specific data aggregation tasks in semiconductor chip test data analytics. Each function accepts input data (typically in the form of a pandas DataFrame) along with parameters that specify grouping fields, identifiers (such as lot IDs), or columns of interest, and returns aggregated results suitable for further analysis or reporting.