NIP Data Models and Flow

Document Number ..... SKA-TEL-SDP-00000??
Context ..... SDP Work Package
Revision 1B
Author ..... R. J. Lyon, B. W. Stappers, E. D. Barr, L. Levin-Preston
Release Date 2016-09-05
Document Classification ..... Unrestricted
Status ..... Draft

Lead Author	Designation	Affiliation
R. J. Lyon	Designation - to be filled in	University of Manchester
Signature & Date:		

Owned by	Designation	Affiliation
Name of the owner - to be filled in	Designation - to be filled in	Affiliation - to be filled in
Signature & Date:		

Approved by	Designation	Affiliation
Name of the approver - to be filled in	Designation - to be filled in	Affiliation - to be filled in
Signature & Date:		

Released by	Designation	Affiliation
Paul Alexander	SDP Project Lead	University of Cambridge
Signature & Date:		

Revision	**Date of issue**	**Prepared by**	**Comments**
1B			

ORGANISATION DETAILS

Name	Science Data Processor Consortium
Address	Astrophysics Cavendish Laboratory JJ Thomson Avenue Cambridge CB3 0HE
Website	http://ska-sdp.org
Email	ska-sdp-pa@mrao.cam.ac.uk

Table of Contents

List of abbreviations

CSP	Central Signal Processor
DFMDL	Data Flow Model
DEC	Declination
DM	Dispersion measure
DO	Data Object
DP	Double Precision Floating Point
FLOPS	Floating point Operations
IO	Input / Output
INT	Intermediary
LMC	Local Monitoring & Control (SDP Work Package)
LSM	Local Sky Model
MSP	Millisecond pulsar
NIP	None Imaging Pipeline
OCLD	Optimal Candidate List and Data
OCL	Optimal Candidate List
PIP	Pipelines (SDP Work Package)
PSRC	Pulsar candidate
PSRFITS	A standard FITS-based format for pulsar data files
PTA	Pulsar timing array
RA	Right Ascension
RFI	Radio Frequency Interference
SaDT	Signal and Data Transport
SDP	Science Data Processor
S/N	Signal-to-noise ratio
SPC	Single pulse candidate
SP	Single Precision Floating Point
TBC	To be confirmed
TBD	To be decided
Timing residual	differences between measured pulse arrival time and expected arrival time
TM	Telescope Manager
TOA	Time-of-arrival

List of symbols

M_{burst}	Size of single pulse data file
M_{cand}	Size of pulsar candidate data cube
M_{input}	Size of input data per observation
M_{metadata}	Size of metadata
M_{obs}	Average data size per observation
$M_{\mathrm{obs;MAX}}$	Maximum data size per observation
M_{output}	Size of output data per observation
M_{pulsar}	Size of pulsar timing data cube
M_{sheet}	Size of an S/N sheet
N_{avg}	Number of averages
N_{beams}	Number of beams simultaneously observed by the array
N_{MAXbeams}	Maximum number of beams simultaneously observed by the array
N_{bin}	Number of bins
$N_{\mathrm{bin:smooth}}$	Number of bins included in smoothing window
N_{bit}	Number of bits
N_{burst}	Number of candidate bursts
N_{byte}	Number of bytes
$N_{\mathrm{byte;sheet}}$	Number of bytes per S/N sheet sample
$N_{\mathrm{byte,pix}}$	Number of bytes per pixel
N_{cand}	Number of pulsar candidates per beam
$N_{\mathrm{ch:smooth}}$	Number of frequency channels included in smoothing window
N_{chan}	Number of channels
N_{FLOPS}	Number of floating point operations per second
N_{FLOPsamp}	Number of floating point operations per sample
$N_{\mathrm{heuristic}}$	Number of heuristics to be extracted
N_{it}	Number of iterations
N_{ops}	Number of operations
N_{param}	Number of parameters
N_{pol}	Number of polarisation products
N_{samp}	Number of samples
N_{source}	Number of sources
N_{sub}	Number of sub-integrations
N_{sum}	Number of sums
N_{unique}	Number of unique candidates
R	Compute rate (FLOPS)
t_{comp}	Computation time
T_{obs}	Observation time
T_{sub}	Sub-integration time
$Trial_{\mathrm{p}}$	Trial periods

$Trial_{\dot{\mathrm{p}}}$	Trial period derivatives
$Trial_{\mathrm{dm}}$	Trial DMs
Δf	Frequency resolution
Δt	Time resolution (sec)

Summary

The purpose of this document is to describe the data types and data flow, between NIP pipeline components identified in the PDR.02.05 Pipelines Element Subsystem Design document (SKA-TEL-SDP-0000027). This includes a description of the data moving through the Pulsar Search Pipeline, Transient Detection Pipeline, and finally the Pulsar Timing Pipeline. This work is a contribution to the T4A-1 “Data Models” task, identified in the SDP work package.

Applicable and reference documents

Applicable Documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, *the applicable documents* shall take precedence.

Reference Number	Reference

Reference Documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, *this document* shall take precedence.

Reference Number	Reference
SKA-TEL-SDP-0000027	PDR.02.05 Pipelines Element Subsystem Design
TEL-SDP-0000040	Parametric models of SDP compute requirements
TEL-SDP-0000080	SDP Memo: SDP Non-Imaging Processing Compute Requirements

1 Overview

1.1 Pulsar Search Pipeline

The CSP will produce a number of pulsar candidates (PSRCs) to be analysed by the SDP. The search pipeline takes these candidates, and filters them producing a subset of pulsar candidates most likely to be of scientific utility. To achieve this the pulsar search pipeline performs multiple filtering operations. The first attempts to identify duplicate detections and those likely arising from RFI, by comparing candidates across multiple telescope beams. The second begins with the determination and extraction of numerical descriptors (known as features or heuristics) for each individual candidate. These describe the important characteristics of the candidate detection. These descriptors are used as inputs to a machine learning system, that will classify each candidate. The outputs of the machine learning system will enable a further filtering of the candidates. Those candidates not considered duplicates or RFI, which the machine learning system also indicates to be astrophysical in origin, generate an alert. However all candidate metadata is retained, including the metadata belonging to those candidates believed to be RFI or noise. This is to allow for later data reprocessing, and to enable the machine learning system to learn from collected data. The latter will facilitate improved filtering performance over time.

1.2 Transient Search Pipeline

The CSP will produce a number of transient single pulse candidates (SPCs) to be analysed by the SDP. The transient search pipeline takes these single pulse candidates, and filters them producing a subset consisting of those most likely to be of scientific utility. This pipeline incorporates the same general processing steps as the pulsar search pipeline. The data is first filtered for duplicates and RFI detections, before being passed through a machine learning system to be assigned labels. The alert generation system is triggered when appropriate. However there are subtle differences between the transient and pulsar pipelines. The transient pipeline will likely extract different heuristics, and the machine learning system used will definitely be different, as the data being processed by the two pipelines is very different.

1.3 Pulsar Timing Pipeline

The CSP will deliver integrated pulse profiles (IPPs) corresponding to known pulsars to the SDP. The job of SDP is to first calibrate and clean the data it receives from CSP, before analysing it with the aim of updating the pulsars timing model. To achieve this the SDP executes multiple processing steps. The first involves performing final detailed flux and polarisation calibration. The calibration is done using external inputs from LMC, and is essential for the high precision timing task. RFI mitigation is also undertaken. Once complete, a number of data products are generated that provide different representations of the data. The subintegrations are combined using the ephemeris contained in the PSRFITS files (Hotan et al., 2004), into one file containing all subintegrations. The TOAs are then determined, the timing residuals computed, and these are used to update the timing model for the pulsar. The different archive products generated by this pipeline are useful for any future post-processing that is likely to occur. The determination of the arrival time of the pulse is also essential as part of observational data quality assurance checks, but also potentially for identifying interesting scientific events.

2 Functions & Data Products

2.1 Pulsar Search Pipeline

2.1.1 Input/Output Data & Rate

The search pipeline will receive pulsar candidates from CSP in the form of OCLDs. Each unique candidate in the OCLD is comprised of three components. These are i) a detected folded (stokes) data cube currently assumed to be in PSRFITS format (Hotan et al., 2004), ii) an S/N sheet (three 2-dimensional matrices) which describe the resulting folded S/N ratio for different combinations of folding parameters (period, period derivative and DM trial values) used when folding the data file, and finally iii) associated metadata in ASCII format. Each data cube will be of size $M_{\rm cand} = N_{\rm chan} \times N_{\rm bin} \times N_{\rm sub} \times N_{\rm byte}$. Whilst the size of each S/N sheet is given $M_{\rm sheet} = (Trial_{\rm p} \times Trial_{\dot{\rm p}}) + (Trial_{\rm p} \times Trial_{\rm dm}) + (Trial_{\dot{\rm p}} \times Trial_{\rm dm}) \times N_{byte;sheet}$. Finally the metadata file will be of size $M_{\rm metadata} \approx 10$ kB. Thus an individual candidate is $M_{\rm cand} + M_{\rm sheet} + M_{\rm metadata}$ in size.

Parameter	Symbol	Expected Value
Number of beams	$N_{\rm beams}$	1500 (SKA-Mid); 500 (SKA-Low)
Number of candidates per beam	$N_{\rm cand}$	1000
Number of frequency channels	$N_{\rm chan}$	128
Number of pulse profile bins	$N_{\rm bin}$	128
Number of subintegrations	$N_{\rm sub}$	64
Typical observation length	$T_{\rm obs}$	600 seconds
Number of bytes	$N_{\rm byte}$	1 byte
Number of bytes per S/N sheet sample	$N_{\rm byte;sheet}$	2 byte
Number of trial periods	$Trial_{\rm p}$	256
Number of trial period derivatives	$Trial_{\dot{\rm p}}$	256
Number of trial DMs	$Trial_{\rm dm}$	256

Table 1: Pulsar search parameters

A description of the pulsar search parameters and the corresponding expected values are listed in Table 1 (taken from TEL-SDP-0000080). Using the parameters provided, the total amount of data entering and leaving the pipeline can be estimated. For example for SKA-Mid, each candidate is approximately 1.45 MB in size. This includes a 1,048,576 sample data cube which is approximately 1.049 MB, S/N sheets of 0.393 MB, and 10 kB of metadata. The total volume of pulsar search data entering SDP for SKA-mid is given by $M_{\rm input} = N_{\rm beams} \times N_{\rm cand} \times (M_{\rm cand} + M_{\rm sheet} + M_{\rm metadata}) = 2.17769$TB per observation. This data is assumed to be received over 100 seconds. This is a constraint imposed on CSP, as both transient and pulsar search data have to be sent to SDP within 600 seconds (observation length) to maintain real-time operation. 500 seconds is the time required to trasmit transient data, with the remaining 100 seconds used to transmit pulsar search data. This yields an input data rate to SDP of 174 Gb/s (gigabits per second).

The output from the pulsar search pipeline will consist of metadata files and data cubes for all unique candidates. The total output data size per observation was originally estimated in an SDP memo (TEL-SDP-0000080, SDP Memo: SDP Non-Imaging Processing Compute Requirements). It assumed 10kB of metadata, and that approximately 1 in 10 candidates

would be retained for archival. The output size was then estimated as $M_{\rm output} = N_{\rm unique} \times M_{\rm cand} + N_{\rm beams} \times N_{\rm cand} \times M_{\rm metadata} = 172$ GB. This estimate has been updated to reflect the inclusion of S/N sheets in the OCLD. Again assuming 1 in 10 candidates entering SDP is deemed worthy of permanent archival, the volume of data to be stored is 218 GB per pulsar search observation.

2.1.2 Merge OCLDs

Merge OCLDs Function Definition

Function	Merge OCLDs	**Owner**	SDP
Brief Description	To remove duplicate candidates detected in multiple beams		
Input Parameters	$N_{\rm beams}$, $N_{\rm cand}$, $N_{\rm param}$		
Input Data	OCLD	**Output Data**	SOCLD
Detailed Description:			

Accepts OCLD data from CSP. OCLD's contain metadata which includes a candidate list. The list describes each candidate's spin period, S/N, dispersion measure (DM), Right Ascension (RA) and Declination (DEC), but other parameters may be required (e.g. period derivative and acceleration). 1 OCLD will be delivered to CSP per beam. Each OCLD is expected to contain $N_{\rm cand} \approx 1000$. This is based on the analysis in Lyon et al. (2016), which shows that real pulsar candidates are often found at lower S/N values than noise candidates. Hence many candidates must be saved for a thorough analysis, so that pulsars are not missed. The function accepts these candidates and performs an operation known as 'sifting' or 'coincidence matching'. Sifting is an algorithmic operation similar in complexity to many standard sorting algorithms, i.e. quick sort (Hoare, 1962). However it is a matching procedure. It compares candidate detections across multiple beams, and identifies those most likely to be duplicates (via similarity tests). The function produces a sifted OCLD (SOCLD). This contains the strongest S/N detection of any candidates which do have duplicates. It also contains those candidates for which no duplicates could be identified. Sifting greatly reduces the number of candidates to be processed further along the processing chain.

Merge OCLDs Data Product

Data Product	SOCLD	**Created by**	Merge OCLDs
Brief Description	An OCLD with duplicate candidate detections removed		
Type	INT	**Dimensions**	Varies (detail below).
Detailed Description:			

This will contain candidates stored in the same format as any other OCLD. However it will largely be devoid of weak S/N duplicate detections and strong sources of RFI. It contains the strongest S/N detection of any candidates for which there are duplicates, and those candidates for which no duplicates could be found. The SOCLD will also retain counts allowing us to keep track of duplicates within a beam and across beams. These counts are retained on a per candidate basis (e.g. candidate n has d duplicates in beam b. This information is important for accurate filtering, but also for data quality assurance (high numbers of duplicates may suggest an observation polluted with interference, for example). The dimensions of the sifted OCLD depend entirely on the data received from CSP, which will vary for each observation. For a worst case scenario where sifting is completely ineffective, the sifted OCLD will contain $N_{\mathrm{cand}} \times N_{\mathrm{beam}}$ candidates, along with their metadata and S/N sheets. This corresponds to the case where no duplicates are found. In reality sifting will be effective. It is estimated that the sifted OCLD will contain $N_{\mathrm{unique}} \approx 0.1 \times N_{\mathrm{cand}} \times N_{\mathrm{beam}}$ candidates.

2.1.3 Heuristic Generation

Heuristic Generation Function Definition

Function	Heuristic Generation	**Owner**	SDP
Brief Description	Extracts numerical descriptors from each candidate		
Input Parameters	Heuristics to extract		
Input Data	OCLD	**Output Data**	Heuristic Metadata
Detailed Description:			

Accepts an OCLD or compatible variant (i.e. SOCLD). Determines and extracts numerical values describing the key characteristics of a candidate. This function extracts heuristics from all the N_{unique} candidates stored in the OCLD. Heuristics are computed from components of the standard pulsar candidate. These components include, but are not necessarily limited to, the integrated pulse profile, the sub-band and sub-integration matrices, and the dispersion measure (DM) against signal-to-noise ratio curve (DM-SNR curve). The determination of which heuristics to extract from these components, can be done either dynamically or off-line. In the dynamic case, heuristics are extracted independently of human input using machine learning tools (e.g. neural networks). The most common approach however is to determine the heuristics to use off-line, prior to data processing. The exact heuristics, and therefore the associated algorithms that calculate them, are still a matter of study (see, Eatough et al., 2010; Bates et al., 2012; Morello et al., 2014). Recent work (Lyon et al., 2016) utilises 8 heuristics in total.

Heuristic Generation Data Product

Data Product	Heuristic Metadata	**Created by**	Heuristic Generation
Brief Description	Numerical descriptors describing each pulsar candidate		
Type	INT	**Dimensions**	$N_{\mathrm{unique}} \times N_{\mathrm{heuristic}}$
Detailed Description:			

This meta data will list the computed heuristic values (real numbers) for each candidate. This meta data will be appended to the OCL component of an OCLD during processing. If each individual heuristic is represented using 8 bytes, then the amount of heuristic metadata generated per candidate is given by $N_{\mathrm{heuristic}} \times 8$ bytes. Whilst for an observation the total amount of heuristic metadata generated is given by $N_{\mathrm{unique}} \times N_{\mathrm{heuristic}} \times 8$. For the case where $N_{\mathrm{unique}} = 1000$, and $N_{\mathrm{heuristic}} = 8$, then 64kB is generated in total.

2.1.4 Candidate Classification

Candidate Classification Function Definition

Function	Candidate Classification	**Owner**	SDP
Brief Description	Predicts a likely origin for each candidate		
Input Parameters	Classifier model, learning parameters, training examples		
Input Data	OCLD with Heuristic Metadata	**Output Data**	Classifications
Detailed Description:			

Candidate classification is a automated decision making process. For each candidate in an OCLD for which there is heuristic metadata, candidate classification assigns a predicted class label. The predicted labels indicate the likely origin of each candidate. The labels are used to make candidate selection decisions, and to determine which candidates to prioritise for attention. The ultimate goal of candidate classification is to categorise candidates into useful groups. These include, but are not limited to: pulsar, non-pulsar, RFI, and noise. The categorisation is accomplished using a machine learning algorithm. There are many possible algorithms that could be used (with varying computational complexities) and these are the subject of study (e.g. Morello et al., 2014; Zhu et al., 2014; Lyon, 2015; Lyon et al., 2016). In general the algorithm is a function f that maps a set of input candidates X, to category labels Y, so that $f : X \mapsto Y$. An accurate mapping from X to Y must first be learned using a set of 'fully labelled' training examples X_{train}. This set contains the candidates x_1 to x_n, where each x_i is representable by an m-dimensional vector. This vector is comprised of heuristics such that $x_i = \{x_i^j, \ldots, x_i^m\}$, for $j = 1, ..., m$, and all $x_i^j \in \mathbb{R}$. The set $X_{train} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$, where each class label $y_i \in Y$, describes the mapping. A similarly defined set X_{test} containing a disjoint set of examples, allows the mapping learned to be quantitatively assessed. Machine learning algorithms can be trained either off-line or on-line. The distinction is important. An off-line classifier need only be trained once prior to the start of a survey, however it is unable to adapt over time. An on-line classifier can learn adaptively over time, however it requires a steady stream of new training examples to refine its accuracy. See TEL-SDP-0000080 and SDP Memo: Apache Storm-based SDP Pipeline for more information.

Candidate Classification Data Product

Data Product	Classification Metadata	**Created by**	Candidate Classification
Brief Description	The predicted class labels for each N_{unique} candidates		
Type	INT	**Dimensions**	$N_{\mathrm{unique}} \times 1$
Detailed Description:			
Contains predicted class labels for each candidate. The metadata will be appended to the OCL component of an OCLD. There will be only one label per candidate. If each individual label is represented using 1 byte (giving possible integer labels from 0-255), then the amount of classification metadata generated per candidate is only 1 byte. For an observation the total amount of metadata generated is given by N_{unique} bytes. For the case where $N_{\mathrm{unique}} = 1000$, then 1kB is generated in total.			

2.1.5 Candidate Selection

Candidate Selection Function Definition

Function	Candidate Selection	**Owner**	SDP
Brief Description	Selects candidates for archival and alert generation		
Input Parameters	Selection threshold		
Input Data	OCLD with classification metadata	**Output Data**	OCLD with selection metadata
Detailed Description:			
This is a decision making step undertaken by a mathematical function. It expects OLCD candidates, along with the metadata that describes the outcome of both sifting and classification. Using this information the function will compute the numerical likelihood of each individual candidate being a new pulsar discovery. Those candidates with likelihoods greater than the selection threshold parameter (provided externally, likely by TM), will then be flagged for alert generation. The likelihood score will be appended to the OCLD metadata. The flagged candidates and their metadata will be sent to the Alert Generation function. For those candidates which are selected, all OCLD data and metadata reaching this stage in the pipeline, will also be sent to the data archive for permanent storage. For all other candidates, only the metadata will be sent to the archive. Note that the metadata is retained to enable refinement of the candidate classification function.			

Candidate Selection Data Product

Data Product	OCLD with selection metadata	**Created by**	Candidate Selection
Brief Description	The OCLD data flagged for alert generation		
Type	INT	**Dimensions**	Varies
Detailed Description:			
This OCLD data contains candidates along with the numerical likelihoods created by the Candidate Selection function. The data dimensions will vary for each observation, primarily according to the input data and sifting accuracy. If each individual numerical likelihood is represented using 8 byte real values, then the amount of additional data generated here is exactly 8 bytes per OCLD candidate. Whilst for an observation, the total amount of additional data generated is given by $N_{\mathrm{unique}} \times 8$ bytes. For the case where $N_{\mathrm{unique}} = 1000$, then 8 kB is generated in total.			

2.1.6 Alert Generation

Alert Generation Function Definition

Function	Alert Generation	**Owner**	SDP
Brief Description	Sends alerts to TM for processing		
Input Parameters	Alert text, Alert metadata, Alert threshold		
Input Data	OCLD with metadata	**Output Data**	TBD / VOEvent
Detailed Description:			
Creates an alert to be sent to TM for each candidate identified during candidate selection. Typically only 1 alert will be generated per observation to prevent flooding the alert system, though it is anticipated that between 0 to 10 could be fired. Likely to be linked to the virtual observatory infrastructure. Those candidates which do not generate an alert, but may still be of scientific interest, must be flagged for later review.			

2.2 Fast Transients / Single Pulse Pipeline

2.2.1 Input/Output Data & Rate

The single pulse/fast transient pipeline will receive single pulse candidates from CSP in the form of OCLDs. Each unique candidate in the OCLD is comprised of two components. This includes i) a detected folded data cube currently assumed to be in PSRFITS format (Hotan et al., 2004), and ii) associated metadata in ASCII format. The metadata will contain information such as:

- The beam number of the detection.
- The timestamp corresponding to the detected signal, dedispersed at a particular dispersion measure with respect to a reference frequency channel.
- The dispersion measure of the detection.
- A set of parameters describing the detection filter (for example, a smoothing co- efficient) and the detection significance (for example, signal to noise ratio).

Each data cube will be of size $M_{\rm burst} = N_{\rm samp} \times N_{\rm chan} \times N_{\rm pol} \times N_{\rm bytes}$. The metadata file will be of size $M_{\rm metadata} \approx 10$ kB. Thus an individual single pulse candidate is $M_{\rm burst} + M_{\rm metadata}$ in size.

A description of the single pulse/fast transient search parameters, and the corresponding expected values are listed in Table 2 (taken from TEL-SDP-0000080). Using the parameters provided, the total amount of data entering and leaving the pipeline can be estimated. For example for SKA-Mid, each candidate is approximately 2.631 MB in size. This includes a 2,621,440 sample data cube which is approximately 2.621 MB, and 10 kB of metadata. The total volume of transient data entering SDP for SKA-Mid, can be estimated assuming an event detection rate. We assume $N_{\rm burst} = 1$ are detected in each beam and sub-integration. Given $T_{\rm obs} = 600$, and $T_{\rm sub} = 1$ second, then we have 1 detection per beam each second. This breaks down as 1,500 events, multiplied by 2.631 MB, which yields 4GB of data each second. For $T_{\rm obs} = 600$ seconds this gives a total input data volume of 2.4 TB. More generally assuming $N_{\rm burst}$ are detected in each beam and sub-integration, that corresponds to $M_{\rm input} = N_{\rm burst} \times N_{\rm beams} \times (M_{\rm burst} + M_{\rm metadata})$ of input data to SDP every $T_{\rm sub}$ seconds. Finally The data collected every $T_{\rm sub}$ seconds is assumed to be received by SDP in no more than 2 seconds. This yields an input data rate to SDP of 15.8 Gb/s (gigabits per second). Note that it is possible that the chosen value for $T_{\rm sub}$ could be around 1 to 10 seconds. Making $T_{\rm sub}$ larger increases the data transfer rate from CSP to SDP, simply by virtue of increasing the volume of data to be captured (buffer $T_{\rm sub}$ seconds of data) however the 2 second transfer time to SDP remains unchanged. This is not necessarily a good idea, as it makes the data rate somewhat bursty in nature. For example if we set $T_{\rm sub} = 10$, and increase $N_{\rm burst} = 10$ such that 1 event is still detected every second, then 39.465 GB of data is generated in $T_{\rm sub}$ seconds (this value is reported in TEL-SDP-0000080). Given the maximum 2 second data transfer time, the input data rate to SDP increases to 157.9 Gb/s (gigabits per second).

The output from the transient search pipeline will consist of metadata files and data cubes for all unique candidates. The total output data size per observation was originally estimated in an SDP memo (TEL-SDP-0000080, SDP Memo: SDP Non-Imaging Processing Compute Requirements). It assumed 10kB of metadata, and that approximately $N_{\rm unique} = 20$ bursts

Parameter	Symbol	Expected Value
Number of beams	N_{beams}	1500 (SKA-Mid); 500 (SKA-Low)
Candidate bursts per beam & sub-integration	N_{beam}	1
Number of samples	N_{samp}	640
Number of frequency channels	N_{chan}	1024
Number of polarisations	N_{pol}	4
Subintegration length	T_{sub}	1 second
Typical observation length	T_{obs}	600 seconds
Number of bytes per candidate	N_{byte}	1 byte
Number of bursts detected	N_{burst}	1 per second per beam

Table 2: Fast Transient search parameters

survive coincidence matching every T_{sub} seconds (which are retained for archiving). The output size was then estimated as $M_{\mathrm{output}} = (N_{\mathrm{unique}} \times M_{\mathrm{burst}} + N_{\mathrm{beams}} \times N_{\mathrm{burst}} \times M_{\mathrm{metadata}}) \times T_{\mathrm{obs}}/T_{\mathrm{sub}} = 4.057$ GB.

2.2.2 Merge OCLDs

Merge OCLDs Function Definition

Function	Merge OCLDs	**Owner**	SDP
Brief Description	To remove duplicate candidates detected in multiple beams		
Input Parameters	N_{beams}, N_{cand}, N_{param}		
Input Data	OCLD	**Output Data**	SOCLD
Detailed Description:			

Accepts OCLD data from CSP. This data is generated for $N_{\mathrm{beams}} \approx 1500$, and a beam produces N_{cand} candidates. OCLD's also contain a candidate list. It includes details of each candidate's beam number, pulse width, S/N, dispersion measure (DM), Right Ascension (RA) and Declination (DEC), but other parameters may be required. A value of $N_{\mathrm{cand}} \approx 1000$ is expected. The function takes these candidates and performs an operation known as 'sifting' as described in Section 2.1.1. It compares candidate detections across multiple beams, and identifies those most likely to be duplicates (via similarity tests). The function produces a new SOCLD containing the strongest unique candidate detections, e.g. those with the highest S/N or via some other measure of significance. This greatly reduces the number of candidates to be processed. Sifting also helps remove candidates arising from RFI, as strong RFI candidates will likely occur in multiple beams with similar properties.

Merge OCLDs Data Product

Data Product	SOCLD	**Created by**	Merge OCLDs
Brief Description	An OCLD with most duplicate candidate detections removed		
Type	INT	**Dimensions**	Varies (detail below).
Detailed Description:			
This will contain candidates stored in the same format as any other OCLD. However it will largely be devoid of duplicate detections and strong sources of RFI. It contains those candidates for which no duplicates could be found, along with the strongest S/N detection of any candidates for which there are likely duplicates. The dimensions of the sifted OCLD depend entirely on the data received from CSP, which will vary for each observation.			

2.2.3 Heuristic Generation

Heuristic Generation Function Definition

Function	Heuristic Generation	**Owner**	SDP
Brief Description	Extracts numerical descriptors from each candidate		
Input Parameters	Heuristics to extract		
Input Data	OCLD	**Output Data**	Heuristic Metadata
Detailed Description:			
This involves the determination and extraction of numerical values which describe the key characteristics of a single pulse candidate. This function is applied to the N_{unique} candidates stored in the sifted OCLD. The heuristics are extracted from the components of each candidate stored in the OCLD. These include, but are not necessarily limited to, the pulse profile, the sub-band and sub-integration matrices, and the dispersion measure (DM) against signal-to-noise ratio curve (DM-SNR curve). Similarly to pulsar search, the exact heuristics, and therefore the associated algorithms that calculate them, are still a matter of study.			

Heuristic Generation Data Product

Data Product	Heuristic Metadata	**Created by**	Heuristic Generation
Brief Description	Numerical descriptors describing each pulsar candidate		
Type	INT	**Dimensions**	$N_{\mathrm{unique}} \times N_{\mathrm{heuristic}}$
Detailed Description:			
This meta data will list the computed heuristic values (real numbers) for each candidate. This meta data will be appended to the OCL component of an OCLD. If each individual heuristic is represented using 8 bytes, then the amount of heuristic metadata generated per candidate is given by $N_{\mathrm{heuristic}} \times 8$ bytes.			

2.2.4 Candidate Classification

Candidate Classification Function Definition

Function	Candidate Classification	**Owner**	SDP
Brief Description	Predicts a likely origin for each candidate		
Input Parameters	Classifier model, learning parameters, training examples		
Input Data	OCLD with Heuristic Metadata	**Output Data**	Classifications
Detailed Description:			

Candidate classification is a automated decision making process. For each candidate in an OCLD for which there is heuristic metadata, candidate classification assigns a predicted class label. The predicted labels indicate the likely origin of each candidate. The labels are used to make candidate selection decisions, and to determine which candidates to prioritise for attention. The ultimate goal of candidate classification is to categorise candidates into useful groups. These include, but are not limited to: known pulsar, non-pulsar, RFI, noise, and burst. The categorisation is accomplished using a machine learning algorithm similar to pulsar search. Such algorithms are the subject of scientific study (e.g. Devine et al., 2016; Wagstaff et al., 2016).

Candidate Classification Data Product

Data Product	Classification Metadata	**Created by**	Candidate Classification
Brief Description	The predicted class labels for each N_{unique} candidates		
Type	INT	**Dimensions**	$N_{\mathrm{unique}} \times 1$
Detailed Description:			

This meta data will list the predicted class labels (real numbers, integers or even textual descriptors) for each candidate. This meta data will be appended to the OCL component of an OCLD. There will be only one label per candidate. If each individual label is represented using 1 byte (giving possible integer labels from 0-255), then the amount of classification metadata generated per candidate is only 1 byte.

2.2.5 Candidate Selection

Candidate Selection Function Definition

Function	Candidate Selection	**Owner**	SDP
Brief Description	Selects candidates for alert generation		
Input Parameters	Selection threshold		
Input Data	OCLD with metadata	**Output Data**	OCLD with metadata
Detailed Description:			
This is a decision making step undertaken by a mathematical function. It expects OLCD candidates, along with the metadata that describes the outcome of both sifting and classification. Using this information the function will compute the numerical likelihood of each individual candidate being of scientific utility. Those candidates with likelihoods greater than the selection threshold parameter (provided externally, likely by TM), will then be flagged for alert generation. The likelihood score will be appended to the OCLD metadata. The flagged candidates and their metadata will be sent to the Alert Generation function. All OCLD data and metadata will also be sent to the data archive for permanent storage. Note that the metadata is retained to enable refinement of the candidate classification function.			

Candidate Selection Data Product

Data Product	OCLD with metadata	**Created by**	Candidate Selection
Brief Description	The OCLD data flagged for alert generation		
Type	INT	**Dimensions**	Varies
Detailed Description:			
This OCLD data contains only those candidates flagged for alert generation by the Candidate Selection function. The data dimensions will vary for each observation, primarily according to the selection threshold chosen. However the accuracy of all previous filtering steps also greatly affect the data dimensions.			

2.2.6 Alert Generation

Alert Generation Function Definition

Function	Alert Generation	**Owner**	SDP
Brief Description	Sends alerts to TM for processing		
Input Parameters	Alert text, Alert metadata, Alert threshold		
Input Data	OCLD with metadata	**Output Data**	TBD / VOEvent
Detailed Description:			
Creates an alert to be sent to TM for each candidate identified during candidate selection. There is expected to be between 0 and 10 transient/single pulse alerts generated per observation.			

Parameter	Symbol	Ordinary Pulsars MSPs	Expected value	PTAs
Maximum umber of beams	$N_{\rm MAXbeams}$	16	16	16
Average number of beams	$N_{\rm beams}$	16	3	3
Number of frequency channels	$N_{\rm chan}$	512	512	4096
Number of pulse profile bins	$N_{\rm bins}$	2048	2048	2048
Number of subintegrations	$N_{\rm sub}$	180	180	180
Number of polarisations	$N_{\rm pol}$	4	4	4
Number of bytes per pixel	$N_{\rm bytes}$	2	2	2
Typical observation length [seconds]	$T_{\rm obs}$	1800	1800	1800
Data size per pulsar [GB]	$M_{\rm pulsar}$	3.02	6.04	48.32
Average data size per observation [GB]	$M_{\rm obs}$	24.2	18.1	145.0
Maximum data size per observation [GB]	$M_{\rm obs;MAX}$	48.3	96.6	773.1

Table 3: Pulsar timing parameters for pulsar timing arrays (PTAs) and millisecond pulsars (MSPs).

2.3 Pulsar Timing Pipeline

The pulsar timing pipeline will receive coherently dedispersed, detected pulsar data cubes from CSP in PSRFITS format (Hotan et al., 2004). This also includes all associated meta-data. In the requirements it is specified that up to 16 pulsars will be able to be observed simultaneously in high precision-timing mode. This means that the data from the SKA1 Mid, will be formed into 16 tied-array beams. These will point anywhere within the primary beam, although they could also correspond to beams generated from different sub-arrays. These beams will have a bandwidth which is typically equal to the total available bandwidth in each of the SKA1-Mid bands, i.e. 300-1000 MHz. This wide bandwidth will be fed into a timing-specific beam-former. It will channelise the data, down to channels that are a few MHz wide, and sum all available/required dishes together coherently. It will then output a complex, i.e. amplitude and phase, data set to the pulsar timing backend. The CSP will be responsible for performing coherent dedispersion (full correction for dispersion in the interstellar medium) on the data, and for folding it at the known pulsar period. It will also form, and partially calibrate, the polarisation of the data.

A description of the pulsar timing parameters and the corresponding expected values are listed in Table 3. Here the expected values are given for three different observing modes: ordinary pulsars, millisecond pulsars (MSPs) and Pulsar Timing Array (PTA) pulsars. As mentioned above, a maximum of 16 tied-array beams will be available for pulsar timing. Since the beams are independent of each other, a combination of the different observing modes can be used simultaneously. This also implies that each pulsar can be processed completely independent of the others, and hence all 16 pulsars do not need to be processed on the same compute node or even the same compute island. Each input data cube will be of size $M_{\rm pulsar} = N_{\rm bin} \times N_{\rm chan} \times N_{\rm sub} \times N_{\rm pol} \times N_{\rm bytes} + M_{\rm metadata}$.

2.3.1 RFI Mitigation

RFI Mitigation Function Definition

Function	RFI Mitigation	**Owner**	SDP
Brief Description	Removes/cleans data affected by RFI		
Input Parameters	TBD		
Input Data	PSRFITS	**Output Data**	PSRFITS file
Detailed Description:			
The first step involves removing any parts of the received data cube that have been affected by RFI. The exact procedure to mitigate the RFI is still under investigation, but commonly used algorithms removing affected frequency channels ("channel zapping") and time samples ("lawn mowing") using a running median. Channel zapping is carried out on the bandpass of each sub-integration and compared to a median smoothed version of the bandpass. In a similar way, the profile lawn mowing is used to replace RFI-affected phase bins with the local median plus some random noise. Here, each sub-integration, frequency channel and polarisation of the pulse profile is compared to a median smoothed version of the same profile.			

RFI Mitigation Data Product

Data Product	RFI Cleaned PSRFITS file	**Created by**	RFI Mitigation
Brief Description	PSRFITS file with most RFI removed		
Type	INT	**Dimensions**	Given below
Detailed Description:			
The cleaned data cube which has the dimensions $N_{\mathrm{bin}} \times N_{\mathrm{chan}} \times N_{\mathrm{sub}} \times N_{\mathrm{pol}}$.			

2.3.2 Calibration

Calibration Function Definition

Function	Calibration	**Owner**	SDP
Brief Description	Calibrates the observation based on information from TM		
Input Parameters	TBD		
Input Data	PSRFITS	**Output Data**	PSRFITS file
Detailed Description:			
After the data cubes have been cleaned from RFI, they are passed on to the calibration step, which calibrates for flux and polarisation.			

Calibration Data Product

Data Product	PSRFITS file	**Created by**	Calibration
Brief Description	The correctly calibrated data		
Type	INT	**Dimensions**	
Detailed Description:			
The calibrated PSRFITS data.			

2.3.3 Averaging

Averaging Function Definition

Function	Averaging	**Owner**	SDP
Brief Description	Creates 4 additional reduced data products for analysis		
Input Parameters	TBD		
Input Data	PSRFITS	**Output Data**	5 PSRFITS files
Detailed Description:			
Thus far the pipeline has been using full-resolution data cubes, since RFI mitigation and calibration greatly benefit from high resolution data. Subsequent processing steps require higher S/N values rather than high resolution, and therefore this step produces a partly averaged data cube to be used through the remaining steps. This branch also produces three additional versions of averaged data cubes, which will be sent to the archive for storage and saved for future post-processing.			

Averaging Data Product

Data Product	5 PSRFITS files	**Created by**	Averaging
Brief Description	Partly reduced data products for archival storage		
Type	INT	**Dimensions**	Varies
Detailed Description:			

The output consists of the following averaged data cubes in PSRFITS format:

- Summed over the entire freq. band. This has a size given by $N_{\rm bin} \times 1 \times N_{\rm sub} \times N_{\rm pol} \times N_{\rm bytes}$
- Summed over all sub-ints. This has a size given by $N_{\rm bin} \times N_{\rm chan} \times 1 \times N_{\rm pol} \times N_{\rm bytes}$
- Summed over the entire frequency band and all sub-integrations. This has a size given by $N_{\rm bin} \times 1 \times 1 \times N_{\rm pol} \times N_{\rm bytes}$
- Summed over part of the frequency band and part of the sub-integrations.
- Original data (non-averaged).

2.3.4 Generate Archive Products

Generate Archive Products Function Definition

Function	Generate Archive Products	**Owner**	SDP
Brief Description	Sends averaged data products to the archive		
Input Parameters	TBD		
Input Data	5 PSRFITS files	**Output Data**	N/A
Detailed Description:			
Simply sends the 5 PSRFITS files generated during averaging to the archive for permanent storage.			

2.3.5 TOA Determination

TOA Determination Function Definition

Function	TOA Determination	**Owner**	SDP
Brief Description	Determines the time of arrival of the detected pulse		
Input Parameters	Pulsar template		
Input Data	PSRFITS	**Output Data**	TOA list
Detailed Description:			
Accepts the partly averaged data cube created during averaging. This is the data cube averaged in both time and frequency. This reduced cube is used for time-of-arrival (TOA) determination. This function will determine the TOAs by cross correlating the current observation, to a pulsar-specific template provided by TM. The function will generate 1 TOA per sub-integration and frequency channel (these are averaged frequency channels and averaged sub-integrations). These TOAs will be stored as a list. There could be 1 TOA list per averaged data product.			

TOA Determination Data Product

Data Product	TOA list	**Created by**	TOA Determination
Brief Description	A list of pulse arrival times per sub-int and freq. channel + metadata		
Type	INT	**Dimensions**	Varies
Detailed Description:			
The pulse arrival times, 1 per sub-integration and frequency channel. The TOA file will also include metadata associated with the observation which describes the telescope, back-end, frequency used.			

2.3.6 Compute Residuals

Compute Residuals Function Definition

Function	Compute Residuals	**Owner**	SDP
Brief Description	Compare expected vs. observed pulse arrival times		
Input Parameters	Pulsar Par file		
Input Data	PSRFITS	**Output Data**	Residuals
Detailed Description:			
This branch will use the currently best timing model to compute expected arrival times based on the model. It will then compare the expected arrival times to the observed arrival times passed on from TOA Determination, and generate timing residuals as the difference between model and observation, and update the model of the pulsar. Potentially, if requested as part of the observing process, this function must also update the timing module.			

Compute Residuals Data Product

Data Product	Residuals	**Created by**	Compute Residuals
Brief Description	The difference in arrival time for the expected vs. observed pulse		
Type	INT	**Dimensions**	TBD
Detailed Description:			
A list of timing residuals.			

2.3.7 Update Timing Model

Update Timing Model Function Definition

Function	Update Timing Model	**Owner**	SDP
Brief Description	Updates the timing model for the observed pulsar		
Input Parameters	TBD		
Input Data	Residuals	**Output Data**	TBD
Detailed Description:			
Update the timing model of the pulsar.			

2.3.8 Alert Generation

Alert Generation Function Definition

Function	Alert Generation	**Owner**	SDP
Brief Description	Sends alerts to TM for processing		
Input Parameters	Alert text, Alert metadata, Alert threshold		
Input Data	OCLD with metadata	**Output Data**	TBD / VOEvent
Detailed Description:			

Creates an alert to be sent to TM or the VOEvent system. In principle, alerts can be generated at various points in the pulsar timing pipeline. These can be loosely broken down into either scientific alerts, or quality assurance alerts. Scientific alerts are generated when there is a significant change in the expected pulse arrival times for a pulsar, or some deviation from the existing timing model, or a change in the pulse shape or flux which indicates that an interesting scientific event may have occurred. Quality assurance alerts are generated by a similar mechanism to scientific alerts. However these are only generated when i) very unusual system behaviour is observed, or ii) upon request by any of the pulsar timing functions which undertake data verification steps. A quality assurance alert indicates the presence of fault in the observing system, or the parameters used to process the timing data (leading to unusual intermediary or final data products). These quality assurance alerts should be sent to TM for investigation.

References

Bates, S. D., Bailes, M., Barsdell, B. R., Bhat, N. D. R., Burgay, M., Burke-Spolaor, S., Champion, D. J., Coster, P., D'Amico, N., Jameson, A., Johnston, S., Keith, M. J., Kramer, M., Levin, L., Lyne, A., Milia, S., Ng, C., Nietner, C., Possenti, A., Stappers, B., Thornton, D., and van Straten, W.: The High Time Resolution Universe Pulsar Survey - VI. An artificial neural network and timing of 75 pulsars, Monthly Notices of the Royal Astronomical Society, 427, 1052–1065, 2012.

Devine, T. R., Goseva-Popstojanova, K., and McLaughlin, M.: Detection of dispersed radio pulses: a machine learning approach to candidate identification and classification, Monthly Notices of the Royal Astronomical Society, 459, 1519–1532, doi:10.1093/mnras/stw655, 2016.

Eatough, R. P., Molkenthin, N., Kramer, M., Noutsos, A., Keith, M. J., Stappers, B. W., and Lyne, A. G.: Selection of radio pulsar candidates using artificial neural networks, Monthly Notices of the Royal Astronomical Society, 407, 2443–2450, 2010.

Hoare, C. A. R.: Quicksort, The Computer Journal, 5, 10–16, doi:10.1093/comjnl/5.1.10, http://comjnl.oxfordjournals.org/content/5/1/10.abstract, 1962.

Hotan, A. W., van Straten, W., and Manchester, R. N.: PSRCHIVE and PSRFITS: An Open Approach to Radio Pulsar Data Storage and Analysis, Publications of the Astronomical Society of Australia, 21, 302–309, doi:10.1071/AS04022, 2004.

Lyon, R. J.: Why Are Pulsars Hard to Find?, Ph.D. thesis, University of Manchester, School of Computer Science, 2015.

Lyon, R. J., Stappers, B. W., Cooper, S., Brooke, J. M., and Knowles, J. D.: Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach, Monthly Notices of the Royal Astronomical Society, 459, 1104–1123, doi:10.1093/mnras/stw656, 2016.

Morello, V., Barr, E. D., Bailes, M., Flynn, C. M., Keane, E. F., and van Straten, W.: SPINN: a straightforward machine learning solution to the pulsar candidate selection problem, Monthly Notices of the Royal Astronomical Society, 443, 1651–1662, 2014.

Wagstaff, K. L., Tang, B., Thompson, D. R., Khudikyan, S., Wyngaard, J., Deller, A. T., Palaniswamy, D., Tingay, S. J., and Wayth, R. B.: A Machine Learning Classifier for Fast Radio Burst Detection at the VLBA, Publications of the Astronomical Society of Pacific, 128, 084 503, doi:10.1088/1538-3873/128/966/084503, 2016.

Zhu, W. W., Berndsen, A., Madsen, E. C., Tan, M., Stairs, I. H., and Brazier, A. e. a.: Searching for Pulsars Using Image Pattern Recognition, The Astrophysical Journal, 781, 117, 2014.